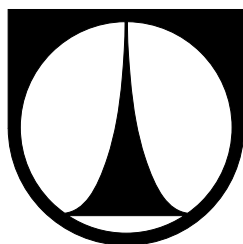


**TECHNICKÁ UNIVERZITA V LIBERCI**  
Fakulta mechatroniky, informatiky a mezioborových studií



## **Samořízené elektrické auto**

Bakalářská práce



**TECHNICKÁ UNIVERZITA V LIBERCI**  
**Fakulta mechatroniky, informatiky a mezioborových studií**

Studijní program: B2646 – Informační technologie

Obor: 1802R007 – Informační technologie

**Samořízené elektrické auto**  
**Self-controlled slot car**

**Bakalářská práce**

Autor práce: **Jakub Vít**

Vedoucí práce: Ing. Jan Koprnický, Ph.D.

Konzultant práce: –

V Liberci 28. prosince 2011

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Jakub Vít**  
Osobní číslo: **M08000187**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Samořízené elektrické auto**  
Zadávající katedra: **Ústav mechatroniky a technické informatiky**

### **Z á s a d y   p r o   v y p r a c o v á n í :**

1. Seznamte se s částmi elektronického systému automaticky řízeného elektrického auta na autodráhu.
2. Realizujte hardwarovou část systému.
3. Navrhněte algoritmus řízení a vytvořte odpovídající software.
4. Funkční auto otestujte na závodní autodráze.

Rozsah grafických prací: dle potřeby dokumentace  
Rozsah pracovní zprávy: cca 40–50 stran  
Forma zpracování bakalářské práce: tištěná/elektronická  
Seznam odborné literatury:


- [1] Brejl, M.; Necesany, J.: Student's contest: Self-driven slot car racing. Oct. 2008, ISBN 978-83 60810-14-9, ISSN 1896-7094
- [2] Ďaďo, S.; Kreidel, M.: Senzory a měřicí obvody. Praha : ČVUT, druhé vydání, 1999, ISBN 80-01-02057-6.

Vedoucí bakalářské práce: **Ing. Jan Koprnický, Ph.D.**  
Ústav mechatroniky a technické informatiky

Datum zadání bakalářské práce: **12. září 2011**  
Termín odevzdání bakalářské práce: **18. května 2012**

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
doc. Ing. Petr Tůma, CSc.  
vedoucí ústavu

V Liberci dne 12. září 2011





## Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis



## Poděkování

Na tomto místě bych rád poděkoval především vedoucímu bakalářské práce Ing. Janu Koprnickému, Ph.D. za cenné rady, pomoc a také za podporu při tvorbě této bakalářské práce a během účasti na soutěži Freescale Race Challenge.

Také bych rád vyjádřil své díky mé přítelkyni Lucii Růžičkové za pomoc při konzultaci slohové stránky práce a veškerou podporu, které se mi z její strany během tvorby bakalářské práce dostávalo.



## Abstrakt

Tato bakalářská práce se zabývá zapojením hardware a návrhem řídicího programu pro autonomně řízené autíčko na autodráhu. Popisuje funkci jednotlivých komponent řídicího obvodu, jehož základním prvkem je mikrokontrolér MCF51JM64. Mikrokontrolér zajišťuje řízení rychlosti jízdy modelu prostřednictvím H-můstku, zároveň dle programu zpracovává data získaná ze snímačů a připojených periférií. Využívá tříosý akcelerometr MMA7361L pro měření přetížení, paměťovou kartu pro ukládání důležitých údajů a reflexní optosnímače CNY70 pro měření ujeté vzdálenosti a snímání povrchu dráhy.

Hlavním bodem práce je návrh a popis algoritmu řízení vozu, který umožňuje modelu jízdu po závodní dráze. Popisuje důležité součásti programu, jejich význam a také jejich funkci v programu.

**Klíčová slova:** Autonomní řízení, Akcelerometr, Freescale Race Challenge, H-můstek, Mikrokontrolér

## Abstract

This thesis deals with the connection of the hardware and the design of control program for autonomously controlled slot car. The thesis also describes the function of individual components of the control circuit where the microcontroller MCF51JM64 is a fundamental element. The microcontroller provides speed control of the model by means of the H-bridge. At the same time, according to the program, it processes the data obtained from sensors and connected peripherals. It uses three-axis accelerometer MMA7361L to measure acceleration, memory card for storing important data, and reflective optical sensors CNY70 for measuring distance and reading the track surface.

The leading point of the thesis is the design and description of the control algorithm which allows the model to ride along the speedway. This algorithm delineates significant components of the program, their importance and their function in the particular program.

**Key words:** Autonomous control, Accelerometer, Freescale Race Challenge, H-bridge, Microcontroller



# Obsah

<b>Prohlášení</b>	<b>5</b>
<b>Poděkování</b>	<b>6</b>
<b>Abstrakt</b>	<b>7</b>
<b>Abstract</b>	<b>7</b>
<b>Obsah</b>	<b>10</b>
<b>Seznam obrázků</b>	<b>11</b>
<b>1 Úvod</b>	<b>12</b>
<b>2 O soutěži Freescale Race Challenge</b>	<b>13</b>
2.1 Důležitá pravidla . . . . .	14
2.2 Poskytnutá podpora . . . . .	14
<b>3 Model automobilu</b>	<b>15</b>
<b>4 Hardware</b>	<b>16</b>
4.1 Mikroprocesor . . . . .	16
4.1.1 GPIO . . . . .	17
4.1.2 KBI . . . . .	18
4.2 Akcelerometr . . . . .	19
4.2.1 Zapojení v obvodu . . . . .	20
4.2.2 Princip funkce . . . . .	20
4.2.3 Výrobní technologie . . . . .	21
4.3 H-můstek . . . . .	21
4.4 Pulsně-šířková modulace . . . . .	22
<b>5 Vývojové prostředí</b>	<b>23</b>



<b>6</b>	<b>Řídicí algoritmus</b>	<b>24</b>
6.1	Činnost algoritmu . . . . .	25
6.2	Tvorba mapy trati . . . . .	27
6.2.1	Zpracování mapy . . . . .	28
6.2.2	Kvalita mapování . . . . .	29
6.3	Závodní režim . . . . .	29
6.3.1	Jízda zatáčkou . . . . .	31
6.3.2	Jízda rovinkou . . . . .	32
6.4	Řešení vypadnutí z dráhy . . . . .	33
6.4.1	Varianta první . . . . .	33
6.4.2	Varianta druhá . . . . .	34
6.4.3	Varianta třetí . . . . .	34
6.4.4	Varianta čtvrtá . . . . .	34
6.5	Řízení rychlosti . . . . .	35
6.5.1	Pasivní brzdění . . . . .	36
6.5.2	Aktivní brzdění . . . . .	36
6.6	Ukládání dat na paměťovou kartu . . . . .	37
<b>7</b>	<b>Nadstandardní HW výbava</b>	<b>40</b>
7.1	Snímač ujeté vzdálenosti . . . . .	40
7.1.1	Obsluha měření ujeté vzdálenosti . . . . .	42
7.1.2	Přesnost měření . . . . .	42
7.2	Snímače startovní čáry . . . . .	43
7.2.1	Nepřesnost snímání . . . . .	44
7.2.2	Obsluha snímání . . . . .	45
7.3	Tvarování signálu snímačů . . . . .	45
<b>8</b>	<b>Problémy</b>	<b>46</b>
8.1	Adheze . . . . .	47
8.2	Akcelerometr . . . . .	47
8.3	Nastavení před jízdou . . . . .	48



<b>Závěr</b>	<b>49</b>
<b>Literatura</b>	<b>51</b>
<b>A Přílohy</b>	<b>52</b>
A.1 Návrh desky pro připojení snímačů . . . . .	52
A.2 Model vozu . . . . .	53
A.3 Schéma řídicího obvodu . . . . .	54
A.4 Obsah přiloženého CD . . . . .	55



## Seznam obrázků

1	Diagram činnosti algoritmu. . . . .	26
2	Trat' ideálně rozdělená na jednotlivé bloky. . . . .	29
3	Diagram činnosti algoritmu během jízdy dle naměřených dat. . . . .	31
4	Startovní čára, křížení a model s vyznačenými snímači trati. . . . .	35
5	Snímač otáček a reflexní značky na kole. . . . .	41
6	Snímače startovní čáry a jejich umístění. . . . .	44
7	Zapojení tvarovacího obvodu při použití tří snímačů. . . . .	47
8	Návrh SMD desky pro připojení čtveřice snímačů. . . . .	52
9	Schéma plošného spoje s obvodem pro tvarování signálu. . . . .	52
10	Pohled na vnitřní vybavení modelu. . . . .	53
11	Kompletní model vozu. . . . .	53
12	Schéma zapojení řídicího obvodu [5]. . . . .	54



# 1 Úvod

Cílem této bakalářské práce je seznámit se s pomocí poskytnuté podpory s konstrukcí autonomního autodráhového modelu vozu. Zkompletovat potřebné prvky obvodu, pochopit funkci jednotlivých hardwarových komponent, zabudovat hardware do modelu a seznámit se se způsobem programování mikrokontroléru a vhodným vývojovým prostředím. Dále pak především vytvořit algoritmus, který bude auto řídit a otestovat jej na závodní dráze.

Model vozu spolu s jeho řídicím algoritmem se také bude účastnit soutěže Freescale Race Challenge. Jedná se o soutěž autonomních autodráhových vozů, které spolu soupeří v rychlosti průjezdu předem neznámé dráhy.

Práce bude členěna na teoretickou a praktickou část. Teoretická část práce bude nejprve osvětlovat princip soutěže a některá důležitá pravidla. Bude také představen použitý model automobilu. Dále se bude zaměřovat na popis jednotlivých elektronických komponent, které zajišťují funkci modelu po stránce hardwaru. Bude vysvětlovat roli jednotlivých komponent v obvodu, případně jejich důležité vlastnosti a parametry, vztahující se k potřebám soutěže. Zběžně také vysvětlí způsob vytvoření řídicího programu ve vývojovém prostředí a jeho nahrání do paměti mikrokontroléru.

Důležitá je především praktická část, jejímž hlavním cílem je kompletace a zabudování hardware do modelu, návrh a vytvoření řídicího algoritmu a také vybavení auta potřebnými dodatečnými komponenty. Součástí následujícího textu je také dokumentace praktické části.





## 2 O soutěži Freescale Race Challenge

Soutěž Freescale Race Challenge, vyhlášená společností Freescale semiconductor probíhá již několikátým rokem. Sdružuje studenty nejrozličnějších elektrotechnických vysokých škol, aby vzájemně porovnali výsledky řešení zadaného problému na závodní trati.

Prvotním úkolem je obdržení hardware zapojit do autíčka namísto původního plošného spoje a upravit karoserii vozu takovým způsobem, aby bylo možné nahrávat řídicí software, pracovat s paměťovou kartou, popřípadě využívat dalších prvků dle potřeby. Následuje návrh, naprogramování a testování řídicího algoritmu v poskytnutém vývojovém prostředí tak, aby bylo autíčko schopné samostatně jezdit po neznámé dráze. Mělo by tedy dokázat prozkoumat dráhu a patřičně na vhodných místech reagovat na její tvar přibrzdováním a zrychlováním.

Kolejnice autodráhy jsou připojeny ke zdroji konstantního napětí, jímž je elektronika autíčka napájena. Po připojení napětí do obvodu se autíčko rozjede. V průběhu prvního kola jízdy neustále snímá dráhu zabudovaným tříosým akcelerometrem. Parametry dráhy poté vhodně ukládá pro následné další využití, například na paměťovou kartu. Ze získaných údajů poté vyhodnotí co možná nejvíce optimální způsob jízdy v dalších kolech.

Vlastní soutěž je rozdělena na hlavní závod a vyřazovací turnaj. Hlavní závod se jede na neznámé trati sestavené ze standardních dílů autodráhy Carrera Evolution na 10 + 10 kol. To znamená, že autíčko v prvním závodě projede deset kol vnitřní stopou dráhy a v druhém závodě deset kol vnější stopou. Pořadí jízdy jednotlivých závodníků v prvním závodě je náhodné, startovní pořadí druhého závodu je od nejpomalejšího vozu. Časy zajeté v obou závodech se poté sečtou a tvoří tak výsledný čas jízdy autíčka.

Vyřazovací turnaj jedou dvě autíčka společně na předem známé symetrické trase<sup>1</sup>. Příklad symetrické trati je vidět viz obr. 2/s. 29. První autíčko z dvojice, které úspěšně dokončí osm kol, pokračuje do dalšího kola vyřazovacího turnaje. Srazí-li se autíčka v místě překřížení trati, automaticky je za vítěze považován vůz, který byl před kolizí ve vedení.

Výhodou pro soutěžní ročník 2011 je možnost auto před závodem testovat a případně změnit některé jízdní parametry programu s ohledem na stav dráhy. Ten určuje například míra jejího opotřebení a stav pneumatik vozu. V následující části jsou zmíněna důležitá pravidla soutěže citovaná z literatury viz [6].

<sup>1</sup>Délka a tvar obou drah jsou totožné.



## 2.1 Důležitá pravidla

- „Pokud některé auto vypadne, soutěžící jej smí znovu nasadit, a to viditelně před místo opuštění dráhy. Pouze jeden člen týmu smí nasazovat auto.“
- „Přítlačné magnety musí být odmontovány, žádné přítlačné magnety nesmí být použity.“
- „Originální podvozek, karoserie, vodící díl, motor a pneumatiky auta Carrera Evolution Audi R8 musí být použity.“
- „Váha auta nesmí překročit 125 gramů.“
- „Auto může být vybaveno jakoukoliv elektronikou, ale pouze uvnitř karoserie.“
- „Na autě může být pouze jeden přepínač, umožňující vybrat jeden ze dvou módů řídicího programu.“

## 2.2 Poskytnutá podpora

Společnost Freescale poskytla všem závodníkům podporu pro snadnou účast v soutěži. A to takovou, aby se i člověk, který nemá velké zkušenosti s programováním mikrokontroléru mohl zúčastnit. Každý závodník obdržel desku plošného spoje se součástkami, a také návody a datasheety k jednotlivým součástkám. Součástí balíčku byla také základní verze programu oživující desku plošného spoje. Program ukazoval způsob, jakým načítat data z periférií či nastavit napětí na motoru.

Bylo tedy nutné prostudovat všechny tyto poskytnuté zdroje, seznámit se s programováním mikrokontroléru a porozumět vlastnostem a roli jednotlivých součástek v systému. Hlavním úkolem však bylo vytvoření řídicího algoritmu, jeho testování na dráze a účast na soutěži.

S přihlédnutím k průběhu minulých ročníků soutěže Freescale Race Challenge se předpokládalo, že všichni účastníci obdrží pouze neosazenou desku plošného spoje a některé součástky. Nicméně ročník soutěže 2011 byl jiný, každý závodník totiž získal plošný spoj již osazený součástkami. K této změně došlo z důvodu zpoždění ve výrobě plošných spojů, čímž se zkrátila doba na přípravu modelu k závodu.



### 3 Model automobilu

Použitý model je standardním vozem řady Evolution z nabídky firmy Carrera. Přesně se jedná o Audi R8 v měřítku 1:32 v bílé barvě viz obr. 11/s. 53. Skládá se z několika dílů a lze jej velice snadno rozebrat. To samozřejmě usnadňuje pravidelné čištění vozu a také provádění úpravy na jeho jednotlivých součástech. Hlavními komponenty vozu jsou: plastový výlisek karoserie, vnitřní část s řidičem, deska plošného spoje, blok s motorem, převodovkou a zadní nápravou, přední náprava s vodítkem a sběrné kartáčky. Již zmíněná modularita konstrukce také umožňuje v případě potřeby poměrně jednoduchou záměnu jízdou opotřebovaných dílů za nové. Jmenovitě se jedná nejčastěji o pneumatiky, sběrné kartáčky a vodítko. Tyto díly nejvíce podléhají opotřebení vlivem častého testování na dráze.

Při zástavbě nového plošného spoje do automobilu bylo pamatováno na možnost snadno připojovat další prvky do obvodu. Všechny vstupy, stejně tak vodiče k dodatečným zařízením bylo třeba osadit konektory a vhodnými protikusy. Je tedy velmi snadné je připojit, odpojit, či v případě potřeby nahradit jinými. Lze tak jednoduše zkoušet různé varianty čidel a snímačů, ale také může na jednom voze pracovat více lidí s tím, že pouze zamění připojené periferie.

Na druhou stranu mělo použití konektorů za následek zvětšení prostorové náročnosti celého zapojení a také nárůst hmotnosti vozu. Karoserie tedy musela být upravena vyjmutím vnitřního plastového dílu s řidičem, díky čemuž byl dodatečně získán potřebný vnitřní prostor. Tato úprava nijak nekolidovala s pravidly soutěže, ba naopak byla schválena jako řešení problému s nedostatkem místa.

Do karoserie bylo také nutné vytvořit dva otvory. Jeden umožňuje vkládat a vyjímat paměťovou kartu, druhý připojovat auto k počítači prostřednictvím USB kabelu. Plošný spoj byl také vybaven dvoupolohovým přepínačem. Jeho přepnutím je možné volit například mezi dvěma režimy chodu programu.



## 4 Hardware

V této části práce bude zběžně popsána funkce důležitých součástí obvodu, jehož schéma je vidět viz obr. 12/s. 54. Důležité je jistě vysvětlit, jak je obvod napájen. Sběrné kartáčky přivádějí přesné napětí trati s hodnotou  $14,8\text{ V}$  na vstup můstkového usměrňovače, který zabezpečuje správnou polaritu napájení obvodu nezávisle na směru jízdy. Napětí dráhy po usměrnění je přímo spínáno H-můstkem a slouží k řízení stejnosměrného motorku automobilu. Protože ostatní elektronika obvodu vyžaduje jinou hodnotu napájecího napětí, je použit napěťový regulátor *LP2951*. Jeho výstupní napětí o hodnotě  $3,3\text{ V}$  spolehlivě zajišťuje napájení zbylých elektronických komponent z kolejnic dráhy a také z USB portu při nahrávání programu.[10]

Na výstup z regulátoru je třeba připojit elektrolytický kondenzátor, kvůli pokrytí výpadků napájení během jízdy. Minimální velikost kapacity kondenzátoru je závislá na množství dodatečně připojeného hardware, který zvyšuje celkový odběr obvodu. Při konfiguraci vozu, která je v práci zmíněna, se osvědčil kondenzátor s kapacitou  $2200\text{ }\mu\text{F}$ .

Pro potřeby ladění programu bylo využito čtyř diod, připojených na výstupní piny procesoru, reprezentujících přední a zadní světla modelu. Na desce je i několik dalších prvků. Lze uvést slot pro paměťovou kartu využívající čtyři piny procesoru, externí oscilátor a mini USB port.

### 4.1 Mikroprocesor

Řízení modelu zajišťuje mikroprocesor MCF51JM64 řady ColdFire V1. Výrobcem je společnost Freescale Semiconductor. Tento nízkopříkonový 32b procesor v pouzdře s 64 piny je taktován externím oscilátorem na  $48\text{ MHz}$ , přičemž maximum taktu je  $50,33\text{ MHz}$  a sběrnice pracuje s frekvencí  $24\text{ MHz}$ . Disponuje  $16\text{ kB}$  operační pamětí a  $64\text{ kB}$  paměti flash pro uložení konstant, proměnných a vlastního zkompilevaného kódu programu. Hodnoty napájecího napětí se pohybují v rozmezí  $2,7\text{ V} - 5,5\text{ V}$ . Tato kapitola a stejně tak její podkapitoly využívají poznatky z literatury viz [4].

Obvod v sobě integruje celou řadu zajímavých komponent umožňujících využití různých způsobů sběru, zpracování a přenosu informací. Mezi využívané lze zařadit například  $12\text{ b}$  analogově-digitální převodník. Ten slouží k převodu analogové veličiny do digitální po-



doby tak, že dokáže přiřadit každému okamžiku vzorkovaného analogového signálu jednu z  $2^{12}$  možných hodnot. Z dalších lze zmínit podporu USB<sup>2</sup> rozhraní, SPI<sup>3</sup> pro komunikaci s pamětovou kartou a časovačů. Ty lze kupříkladu použít pro jakoukoliv plánovací činnost, generování přerušení a PWM<sup>4</sup> signálu pro řízení motoru.

Procesor má velké množství konfigurovatelných vstupně-výstupních portů. Umožňuje snadno přistupovat k jejich hodnotám a také k hodnotám jednotlivých jejich pinů. Některé porty, nebo jen jejich piny, jsou sdíleny s integrovanými zařízeními jako časovače, sériové komunikační rozhraní, nebo třeba A/D převodník. Řada z nich je volně vyvedena na plošném spoji a lze jich tedy snadno využít pro připojení libovolných dalších zařízení, či snímačů. Pro připojení digitálních periférií se v této práci nabízí využití pinů konfigurovaných jako vstupy KBI<sup>5</sup>, nebo GPIO<sup>6</sup>.

Na desce je také vyvedeno rozhraní pro debugging a nahrávání programu bez nutnosti použití bootloaderu. Ve schématu viz obr. 12/s. 54 je připojeno na konektor *CON9*.

#### 4.1.1 GPIO

Nejsou-li piny procesoru využity jinými perifériemi, chovají se jako GPIO. Tyto víceúčelové piny představují jednu z možností, jak velmi snadno připojit k procesoru další zařízení a umožnit jeho použití v mnoha aplikacích. Důležité parametry pinů, ovlivňující jejich chování, jsou nastavitelné pomocí celé řady ovládacích registrů, přidružených vždy k patřičnému portu. Proto chceme-li je využívat, je třeba nejprve nastavit příslušné registry. Základním z nich jsou registry *PTxDDn* nastavující směr toku dat. Určují, zdali bude pin vstupem, či výstupem.

Zamýšlíme-li piny využít jako výstupy, je možné zapnout kontrolu rychlosti přeběhu (Slew Rate) registrem *PTxSEn*. Ta omezuje rychlost změny hodnoty výstupního signálu, udává se ve  $V/\mu s$ . Pokud je kontrola aktivována, je rychlost přeběhu omezena s ohledem na snížení elektromagnetického vyzařování obvodu.

---

<sup>2</sup>Universal Serial Bus

<sup>3</sup>Serial Peripheral Interface

<sup>4</sup>Pulse-width modulation

<sup>5</sup>Keyboard interrupt

<sup>6</sup>General purpose input/output



Také existuje možnost nastavení jedné ze dvou hodnot síly signálu (Drive Strength) s pomocí registru *PTxDSn*. To umožňuje výstupnímu pinu generovat signál o vyšší proudové hodnotě. Díky tomu je následně možné spínat větší zátěž se stejnou rychlostí, jako v případě malého výstupního proudu. Je ale samozřejmě nutné kontrolovat, že nejsou překročeny limity pro celkové maximální proudové zatížení mikroprocesoru. Zvýšením výstupního proudu pinu může také dojít ke zvýšení elektromagnetického vyzařování obvodu.

Jsou-li porty nastaveny jako logické vstupy, ať už víceúčelové, nebo k vnitřním zařízením, lze změnou hodnoty registru *PTxPEn* programově nastavit mají-li na ně být připojeny pull-up odpory. Díky nim lze na pinech zajistit trvalou hodnotu signálu, tedy logickou jedničku. Tímto způsobem lze omezit problémy s neurčitými stavy na vstupech. Po restartu procesoru jsou všechny vstupy nastaveny do stavu vysoké impedance, tváří se odpojeny od sběrnice, nereagují na vstupní signál a pull-up odpory jsou aktivovány.

#### 4.1.2 KBI

Některé z pinů procesoru (na portech B, D a G) lze nastavit jako vstupy pro přerušení z klávesnice. Jsou-li piny nastaveny jako KBI, vyvolá na ně patřičná přivedená logická hodnota přerušení programu. Během přerušení lze obsloužit libovolný kód, obvyklé je však pouze nastavení patřičného příznaku vzniklého přerušení. V programu je poté příznak kontrolován a svědčí-li o nastalém přerušení, je vykonán příslušný kód a pak dojde k vynulování příznaku. Důležité je během obsluhy zakázat reakci na další přerušení. Dojde tak k omezení snímání falešných přerušení.

Aby bylo možno piny v tomto režimu používat, je nejprve třeba provést jejich nastavení. Registr *KBIxPE* náležící k patřičnému portu aktivuje na zvolených pinech přerušení z klávesnice. Je také nutné nastavit registr *KBIxES* určující, zdali reagovat na náběžnou, či sestupnou hranu signálu, nebo na logickou úroveň signálu, případně na vybraných vstupech nastavit použití interních pull-up odporů. Jako další musíme vynulovat příznak přerušení *KBIxSC\_KBACK* a povolit na vybraných pinech přerušení skrze *KBIxSC\_KBIE*. Tím je v podstatě veškeré nastavení dokončeno a lze jej využívat.



## 4.2 Akcelerometr

Hlavním prostředkem pro snímání jízdních podmínek na trati je obvod tříosého akcelerometru MMA7361L. Tento mikromechanický kapacitní snímač, vyrobený technologií MEMS<sup>7</sup>, je předurčen pro jednoduché použití v bateriemi napájených malých elektronických zařízeních vyžadujících nízkou spotřebou energie a malé rozměry. Kapitola věnující se akcelerometru včetně podkapitol čerpá z literatury viz [3].

Jeho funkcí v tomto obvodu je snímat přetížení, které působí na model auta v jednotlivých osách. Ať již jde o gravitační zrychlení planety Země, působící stále silou 1 *g* směrem ke středu planety nebo o příčné a podélné přetížení způsobené zatáčením, zrychlováním a zpomalováním vozu na autodráze.

Akcelerometr je jednou z nejdůležitějších součástí, která je v obvodu použita. Představuje „oči“ automobilu a zprostředkovává tak programu spojení s okolním světem. S jeho pomocí je možné snímat během jízdy cenná data o tvaru autodráhy díky přetížení, které na model automobilu při jízdě působí. Nabízí také možnost nastavení několika parametrů v závislosti na způsobu momentálního použití.

Při jeho využití v bateriově napájených zařízeních, tedy v těch, u kterých je požadována nízká spotřeba energie, nabízí akcelerometr takzvaný *sleep mode*. K jeho aktivaci dochází po přivedení nízké logické hodnoty na pin číslo sedm. Poté dojde k vypnutí výstupních pinů obvodu a taktéž ke značnému snížení spotřeby. Aktivní sleep mód zaručuje spotřebu obvodu rovnu přibližně 3  $\mu\text{A}$ . Přivedením vysoké logické hodnoty přejde součástka opět do normálního pracovního režimu.

Součástku lze také nastavit pro měření v jednom ze dvou možných rozsahů dle logické hodnoty přivedené na pin deset, tedy *G-Select*. Lze tedy vybrat mezi rozsahy 1,5 *g*, či 6 *g*. To je ideální, požadujeme-li pro optimální funkci obvodu dvě měřicí citlivosti. Změna rozsahu může být provedena kdykoliv během činnosti obvodu. Postačuje-li rozsah 1,5 *g* je možné nechat tento pin nezapojený a o nastavení se postará obvod sám.

Akcelerometr disponuje schopností *self test*, umožňující otestovat správnou funkčnost svých vnitřních elektrických a mechanických komponent a tím pádem také například zjistit, byl-li správně připojen do obvodu. Je-li zahájen test, začne na každou z os akcelerometru

---

<sup>7</sup>Micro-Electro-Mechanical-Systems



působit elektrostatická síla, která způsobí její drobné vychýlení. S osami  $x$  a  $y$  je mírně vychylováno, zatímco na ose  $z$  je nastaveno zrychlení rovnající se  $1\text{ g}$ . Tato schopnost se hodí hlavně v aplikacích, kde je nutné kontrolovat správnou funkčnost součástky po celou dobu její životnosti, například v detektorech pádu a náklonu.

Integrovaný akcelerometr může být například použit pro měření odstředivé síly, zrychlení, náklonu, rychlosti. Taktéž pro snímání pohybu a polohy v ovladačích zábavní elektroniky a mobilních telefonech, a také pro detekci pádu v citlivých zařízeních a přístrojích jakými jsou třeba notebooky s klasickými pevnými disky.

#### 4.2.1 Zapojení v obvodu

Zapojení akcelerometru v obvodu je velmi jednoduché. Zaprvé je k patřičným pinům součástky přivedeno napájecí napětí. Dále je na pin sedm, zprostředkovávající sleep mode, přivedena logická jednička proto, aby byl akcelerometr neustále v činnosti. Vzhledem k napájení z kolejnic autodráhy je možnost spořit energii vypínáním akcelerometru zbytečná. Pin deset, jenž určuje měřicí rozsah součástky, je připojen na zem. Rozsah měření je tím nastaven na  $1,5\text{ g}$ . Posledními využitými piny obvodu jsou výstupy, jejichž hodnota signálu odpovídá velikosti působícího přetížení v jednotlivých osách. Tyto vodiče jsou připojeny na vstupy analogově-digitálního převodníku procesoru. Program díky převodníku pravidelně snímá výstupy akcelerometru a získává tak absolutní hodnotu analogového signálu v každém okamžiku měření. Tyto nasnímané hodnoty ještě zpracovává filtrem a následně poskytuje jejich zpracovanou hodnotu k dalšímu použití.

#### 4.2.2 Princip funkce

Zařízení se skládá z kapacitního senzoru (g-cell) a obvodů pro další zpracování signálu. Kapacitní senzor je mechanická struktura utvořená z polovodičového materiálu. Lze si ji představit jako pohyblivý prvek upevněný na nosnících. Tyto nosníky se pod vlivem přetížení, jemuž je obvod vystaven, deformují a deformace je převáděna na změnu kapacity proměnného kondenzátoru. Je tedy využito nelineární závislosti mezi velikostí kapacity kondenzátoru a vzdáleností mezi elektrodami. Čím je vzdálenost mezi elektrodami menší, tím větší je výsledná kapacita kondenzátoru. Integrovaný obvod poté vyhodnotí změnu kapacity každého z kondenzátorů a velmi přesně určí velikosti působícího zrychlení.





### 4.2.3 Výrobní technologie

Technologie MEMS je jednou z mikrosystémových technologií používaných pro tvorbu miniaturních 3D struktur s rozměry v rozmezí 1–1000 mikrometrů. Díky těmto technologiím je možné vytvářet různými mikroobráběcími procesy mikrostruktury použitelné pro výrobu mikrosenzorů (například akcelerometry, gyroskopy, biosenzory, tepelné a optické senzory), mikroaktuátorů a také jednotlivých součástek (mikrorelé, mikrocívky, mikrospínače).[7]

Jejich nespornou výhodou je bezproblémová kompatibilita s klasickými běžně používanými mikroelektronickými technologiemi, což umožňuje integrovat nejrozličnější mikromechanické obvody dohromady na jeden čip s elektronickými obvody, jako jsou například ty pro úpravu a zpracování signálu. Vznikne tak systém mající výhody integrovaných obvodů. Tudíž malé rozměry, spolehlivost, nízkou cenu a zároveň vlastnosti jenž poskytují mikrosenzory.[7]

## 4.3 H-můstek

Model je vybaven malým kartáčovým stejnosměrným motorem s buzením permanentními magnety. Proto aby bylo možné ovládat rychlost jízdy, je nutné měnit napětí na vstupech motoru. Nejjednodušší způsob řízení otáček je možné realizovat pomocí jednokvadrantového regulátoru. Tedy spínání napětí na motoru tranzistorem. Tento regulátor je ale nevhodný především z důvodu vysokých ztrát a nemožnosti změny smyslu otáčení. Dvoukvadrantové řízení tranzistorovým půlmůstkem je také nevhodné. Umožňuje sice měnit otáčky motoru a smysl otáčení, nicméně nelze s jeho pomocí využívat aktivního brzdění. Jako nejvýhodnější se jeví využití čtyřkvadrantového řízení za pomoci můstkového spínače, nejlépe H-můstku se spínáním řízením. Jeho hlavní výhodou jsou velmi nízké ztráty na spínacích tranzistorech.[9]

Obvod je tvořen čtyřmi tranzistory, jejichž sepnutí řídí dva vstupní vodiče  $IN1$  a  $IN2$ . Každý z nich ovládá jednu polovinu můstku. Různými kombinacemi sepnutí tranzistorů je možné na každý z pólů motoru připojit buď napájecí napětí, nebo zem. Z toho vyplývá možnost velmi jednoduše volit směr otáčení motoru, rychlost otáčení, případně nechat motor volně otáčet či aktivně brzdit. Brzdění je rozvedeno dále v kapitole 6.5. Část práce věnující se h-můstku čerpá z literatury [2], kde lze o tomto obvodu nalézt další informace.

H-můstek 33931 je tedy integrovaná součástka pro výkonové řízení induktivních zátěží s proudy se špičkami až 5 A. Maximální velikost proudu samozřejmě závisí na možnostech



chlazení obvodu v dané aplikaci. Je-li na vstup přiveden PWM signál, je výstup obvodu možno spínat s frekvencemi až do 11  $kHz$ . Obvod má také zpětnovazební výstup  $FB$ . Na jeho výstupu je poskytováno 0,24 % z hodnoty výstupního proudu, díky čemuž je možné analogově-digitálním převodníkem monitorovat v reálném čase zátěž motoru.

Další z výstupů můstku značený  $SF$  je určen k indikaci nízké hodnoty napájecího napětí, překročení maximálního proudu a překročení přípustné teploty obvodu. Obvod je vybaven schopností vnitřního omezení výstupního proudu. To probíhá pomocí vhodného automatického zmenšení střidy výstupního PWM signálu tak, aby proud nepřekročil přípustnou hodnotu. Velký výstupní proud má také vliv na růst teploty přechodů tranzistorů, proto je nutné ho omezovat, nebo celý obvod úplně vypnout. Dosáhne-li teplota přechodu 160 °C, dojde k omezení proudu na 4,2 A. Dalším růstem na 175 °C přejde obvod do vypnutého stavu a indikuje na svém výstupu chybu přehřátí.

## 4.4 Pulsně-šířková modulace

Pro efektivní ovládání H-můstku je využito pulsně-šířkově modulovaného signálu. Jeho základem je pevná nosná frekvence a amplituda signálu, zatímco šířka impulsu (střída) je proměnlivá.

Takto modulovaný signál je složený z periodicky se opakujících obdélníkových pulsů proměnlivé šířky, proložených nulovými pulsy (prázdnými místy). Lze si to představit jako ekvivalent rychlého připojování a odpojování napájení. Motor díky své setrvačnosti a induktivnímu charakteru nestíhá sledovat rychlé změny signálu. Chová se tak stejně, jako by byl napájen konstantním napětím. Poměr mezi střídou (dobou trvání) obdélníkových pulsů a prázdnými místy je vyjádřen v procentech. Při poměru 100 % je motor napájen maximální hodnotou napájení, poměr 50 % udává stejnou délku sepnutí i vypnutí napájecího napětí. Při poměru 50 % je střední hodnota signálu rovna polovině maximálního napětí.

Pulsně-šířková modulace tak nabízí velmi snadnou možnost regulace otáček motoru pouhou programovou změnou střidy signálu.



## 5 Vývojové prostředí

Jako vývojové prostředí pro programování mikroprocesoru ColdFire byl použit nástroj Code Warrior ve verzi 6.3. Prostředí je v omezené verzi ke stažení na stránkách výrobce. Pomocí tohoto nástroje je možno celkem jednoduše psát vlastní kód programu, a také jej překládat a vytvářet tak výstupní program srozumitelný mikroprocesoru.

Byl použit programovací jazyk ANSI C, který lze využít k programování celé řady mikroprocesorů a jejich aplikací. Pro programování procesoru je vhodný díky uživatelské přívětivosti a přehlednosti vytvořeného kódu programu při porovnání například s jazykem symbolických adres.

Nahrát program do mikrokontroléru je možné dvěma způsoby. První a díky své jednoduchosti také víceméně jedinou využívanou možností je nahrání programu skrze USB propojení s počítačem. Procesor po připojení kabelu vyhodnotí zdroj a hodnotu napájení a zahájí nahrávání bootloaderu. Ten poté zajistí, že se paměť pro program procesoru stane přímo přístupnou jako běžné paměťové zařízení. Správné naběhnutí bootloaderu a připojení zařízení je signalizováno rozsvícením předních světel modelu a souborem s názvem READY.txt, jenž se jako jediný nachází na připojeném zařízení. Následně již jen stačí nahrát vývojovým prostředím zkompilovaný soubor s příponou .s19. Úspěšné nahrání programu signalizuje opět rozsvícení, tentokrát však zadních světel a změna názvu souboru READY.txt na SUCCESS.txt.

Druhou možností je použít hardwarové rozhraní BDM<sup>8</sup> vyvedené na plošném spoji. To ale prakticky znamená demontáž karoserie auta a případně dalších vnitřních komponent proto, aby byl umožněn přístup k patřičnému konektoru.

Po úspěšném nahrání a položení na dráhu začne procesor okamžitě vykonávat instrukce nahraného programu.

---

<sup>8</sup>Background Debug Mode



## 6 Řídicí algoritmus

Algoritmus se snaží zajistit dostatečně rychlou, zároveň však dostatečně bezpečnou jízdu. Cílem je tedy především dokončit závod bez vypadnutí auta z dráhy. Doba, po kterou závodník obchází trať, aby znovu nasadil vůz, může být také rovna času, za který auto dokončí celé jedno kolo.

Veškeré nastavované hodnoty, jako způsob brzdění nebo také rychlost průjezdu tratí byly stanovovány experimentálně, na základě testování autíčka na dráze za různých podmínek. Aby mohl program alespoň minimálně plnit požadovanou funkcionalitu, musí splňovat několik základních požadavků. Jako základní nutné schopnosti algoritmu byly stanoveny tyto čtyři body.

Zprv je třeba, aby program v pravidelných intervalech během jízdy snímal veškerá data o parametrech trati. Ukládá všechny důležité informace, které jsou mu prostřednictvím senzorů a snímačů zprostředkovávány, a to takovým způsobem, aby je mohl kdykoliv v případě potřeby znovu načíst k dalšímu použití. Vytváří tedy mapu autodráhy, kterou následně ukládá do souboru na paměťovou kartu. V praxi je auto vybaveno dvoupolohovým přepínačem na podvozku. Je-li přepínač v poloze A, je mapa ukládána a načítána ze souboru *mapaA.txt*. Je-li přepínač v poloze B, je mapa ukládána a načítána ze souboru *mapaB.txt*. Díky tomu lze na paměťovou kartu uložit dva různé soubory s nasnímanými daty a mezi nimi přepnutím přepínače volit. Toho je možné využít pro nasnímání obou drah trati nebo jedné dráhy v obou směrech jízdy, bez nutnosti vyjímát mezi jízdami paměťovou kartu a předchozí záznam mazat.

Zadruhé musí být auto schopné fungovat v závodním režimu. Tento předpokládá využití nasnímaných dat k optimálnímu průjezdu dráhy. Program volí podle aktuální pozice auta na dráze vhodný jízdní režim, tzn. na rovinkách zrychluje, před zatáčkami brzdí apod.

Zatřetí je vyžadováno správně rozeznat, kdy auto zahájí a dokončí celé jedno kolo trati. Díky tomu je možné využít jízdu prvním kolem k mapování trati a další kola projíždět již v závodním režimu. Během závodního režimu je informace o dokončeném kole stejně potřebná, neboť v každém novém kole je nutné číst mapu trati od začátku.

Posledním požadavkem na program je schopnost reagovat na stav, kdy autíčko vypadne z dráhy a následně je znovu usazeno na závodní trať.



Programový kód je cyklicky vykonáván v nekonečné smyčce, z níž jsou volány jednotlivé potřebné funkce zabezpečující jednotlivé úkony. Využívá také několika přerušení pro provádění pravidelných činností, jakými jsou třeba kontroly vstupů a výstupů procesoru, nebo aktualizace hodnot systémových proměnných.

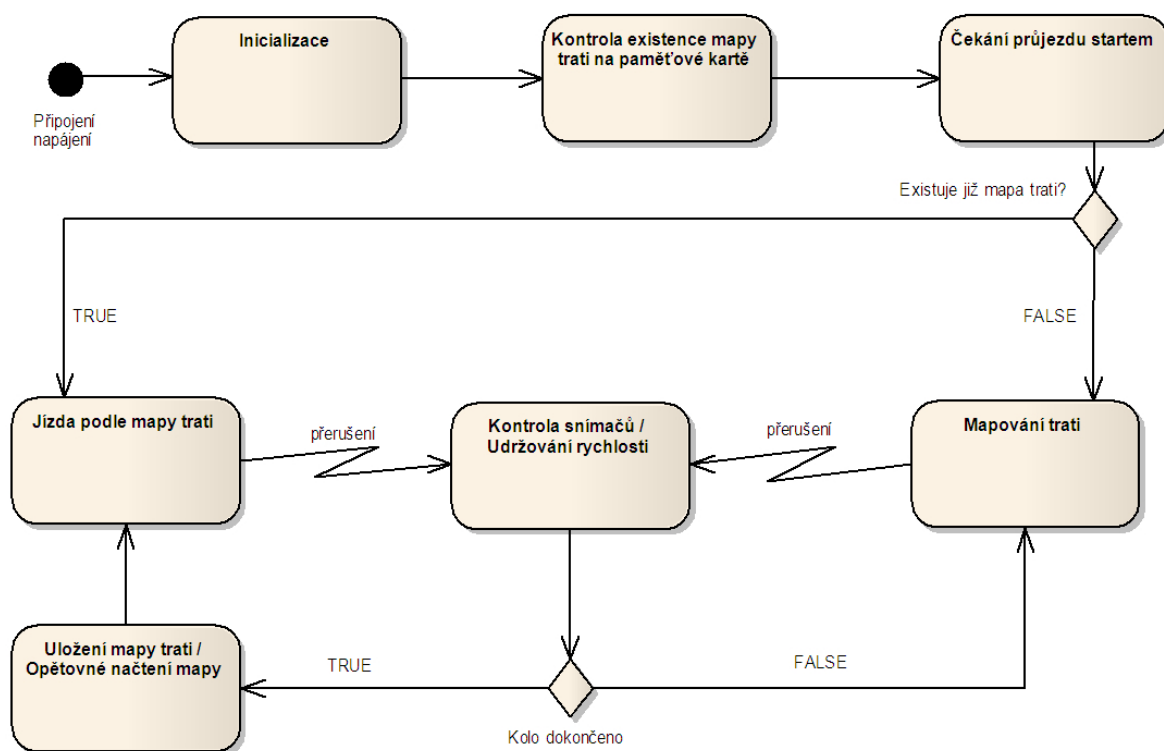
## 6.1 Činnost algoritmu

Vykonávání řídicího programu je zahájeno připojením k napájení prostřednictvím sběrných kartáčků, tedy správným usazením auta na kolejiště autodráhy. Po spuštění nejprve dojde k nastavení parametrů procesoru pomocí řídicích registrů a inicializování všech proměnných.

Spolu se spuštěním programu je vytvořen nový textový soubor, do něhož jsou během jízdy vypisovány potřebné informace. Je tak zajištěna zpětná vazba pro programátora. Jako výhodné se ukázalo vypisování názvů právě prováděných činností programu a programátorem vybraných naměřených a vypočtených hodnot. Kontrolou záznamů v textovém souboru lze odhalit nejrůznější chyby v kódu a také kontrolovat hodnoty, jejichž správnost je zásadní pro chod programu.

Následně je načtena poloha přepínače na podvozku vozu, která udává jméno souboru s mapou. Program zkontroluje datové úložiště na paměťové kartě tak, že se pokusí soubor otevřít. Existence souboru určuje další postup ve vykonávání programu. Existuje-li soubor s nasnímanou mapou trati, algoritmus logicky předpokládá, že již bylo snímání trati v minulosti úspěšně dokončeno a tudíž jej není třeba znovu opakovat. Neexistuje-li soubor, je jasné, že trať musí být nejprve zmapována.

V případě, že se na paměťové kartě nenachází soubor s mapou a v mikrokontroléru je nahrán řídicí program, je postup vykonávání dalších činností následující. Napětí na motoru se zvyšuje, auto se rozjíždí a snaží se udržovat rychlost jízdy dle nastavených hodnot. Mapování závodní dráhy však ještě nezahajuje, čeká totiž na moment, kdy dojde k synchronizaci na autodráze. Tím je myšleno nalezení nějakého vztažného bodu dráhy a tím i uvědomění si své momentální pozice. Z toho důvodu je neustále kontrolován výstupní signál snímačů povrchu trati. Synchronizace nastane při průjezdu startovní čarou, přičemž dojde k zahájení mapování trati 6.2. Jeho ukončení je podmíněno ujetím jednoho celého kola, tedy od startu ke startu.



Obrázek 1: Diagram činnosti algoritmu.

Vytvořená mapa je vhodným způsobem zpracována 6.2.1 a uložena na kartu k dalšímu využití, snímače trati přestávají pracovat a je zahájen režim závodní jízdy 6.3.

Druhou možností je stav, nachází-li se na kartě soubor obsahující zmapované první kolo. V tomto případě zná program tvar a délku trati, ovšem tyto informace lze využít až v okamžiku, kdy zná svou pozici na trati. Z toho důvodu se aktivují snímače povrchu trati a auto se rozjíždí. V momentě kdy nasnímá průjezd startovní čarou je jasné, že se nachází na počátku mapy trati a díky tomu je zahájena závodní jízda. Zjednodušený diagram funkce algoritmu je možné vidět viz obr. 1/s. 26.



## 6.2 Tvorba mapy trati

Mapování trati je souvislý proces zahájený prvním průjezdem přes startovní čáru a končící druhým průjezdem, jehož účelem je zjišťovat a uchovávat parametry závodní trati na základě měření dat poskytovaných připojenými periferiemi. Zde konkrétně akcelerometrem, snímačem ujeté vzdálenosti a snímači povrchu závodní dráhy.

Nejprve je nutné ozřejmit způsob, jakým na dráhu nahlíží řídicí program. Během mapování i v průběhu celé další jízdy je trať chápána jako souhrn jednotlivých bloků typu rovinka, levá zatáčka a pravá zatáčka. Jak by mělo pohledem algoritmu vypadat ideální rozdělení trati po nasnímání, je vidět viz obr. 2/s. 29, kde za blok je považován vždy úsek mezi dvěma červenými čarami.

Všechna potřebná data o namapovaných blocích jsou ukládána do pomocného bufferu. Tím je statické jednorozměrné pole vlastního datového typu *blok\_trati* s parametry *begin*, *type* a *special*. U každého bloku jsou uchovány informace o jeho typu a vzdálenosti, v níž se nachází od startovní čáry. U každého z nich je ještě možno uložit libovolnou celočíselnou hodnotu. K tomu je určen již zmiňovaný parametr *special*. Rozměr pole, tedy počet jeho prvků, je pevně definován na 60 prvků. Toto číslo poskytuje dostatečnou rezervu pro použití na jakékoliv dráze. Počet využitých bloků se po zpracování mapy nejčastěji pohybuje kolem dvanácti až patnácti prvků. Využití statického pole je v tomto případě výhodnější než použití dynamického pole, které sice poskytuje přesně takový datový prostor, jaký je potřeba pro uchování mapy trati, nicméně jeho implementace je náročná na režii ze strany procesoru a způsobila by neúměrný nárůst velikosti kódu programu.

Procesor neustále prostřednictvím analogově-digitálního převodníku kontroluje výstupy akcelerometru a načítá z nich data. Tato jsou pomocí filtrů zpracována a poskytnuta dále pro potřeby programu. Při mapování je kladen důraz na údaje o ose x, tedy o bočním zrychlení. V programu jsou proto definovány hodnoty mezí, s nimiž jsou filtrovaná data porovnávána. Je-li jejich hodnota menší než nižší ze zadaných prahových hodnot, je indikován průjezd vozu pravotočivou zatáčkou. Je-li hodnota dat větší než vyšší z prahových hodnot, jede vůz levotočivou zatáčkou. Poslední možností je hodnota dat o velikosti pohybující se v rozsahu mezi prahovými hodnotami. Pak je jasné, že model projíždí po rovině. Hodnoty mezí je třeba upravit v závislosti na použitém akcelerometru a výsledcích měření, ne každá součástka má



totiž tytéž vlastnosti a výsledné snímání dráhy může být tímto zkresleno. Změnou nastavení mezi lze měnit citlivost a tím i ovlivňovat množství nasnímaných rovin na úkor zatáček a naopak.

Mapování probíhá tak, že po průjezdu startovní čáry je do prázdné mapy přidán první blok typu rovinka s počátkem ve vzdálenosti nula. V momentě kdy je vyhodnoceno naměřené přetížení jako jedna ze zatáček, dojde k přidání druhého bloku, kde jeho počátkem je aktuálně ujetá vzdálenost. Přidání nového bloku do mapy je vždy podmíněno naměřením odlišného typu projížděného úseku trati oproti předchozímu bloku.

Současně dochází během pravidelného přerušení ke kontrolám povrchu trati a čeká se na indikaci průjezdu startovní čarou. Při jejím nalezení je zřejmé, že auto dokončilo právě jedno celé kolo trati. Následuje tedy ukončení mapovacího cyklu.

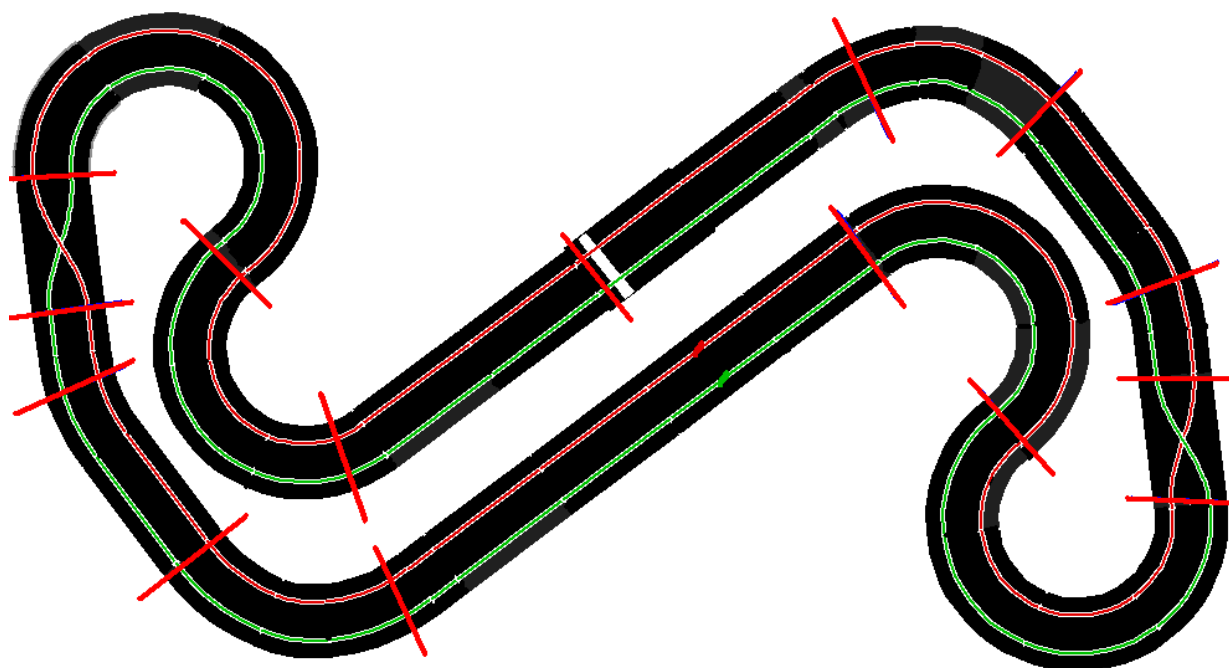
V dalším kroku je přidán poslední blok, jehož úkolem je v očích algoritmu sjednotit cílovou rovinku – tedy dvě rovinky nasnímané před a za cílovou čarou. Program tak „vidí“ při jízdě i za cílovou čáru a poslední projížděný blok během závodu pro něho končí až před první zatáčkou za cílem. Dále je uložena celková délka trati ve formě hodnoty parametru *special* posledního z nasnímaných bloků trati. Díky tomu není třeba ukládat na paměťovou kartu více souborů a všechna potřebná data jsou k dispozici pohromadě v mapě.

Na konec je funkcí *postprocessing\_pole* provedeno zpracování nasnímané mapy trati.

### 6.2.1 Zpracování mapy

Činnost této funkce spočívá v postupném procházení mapy a odstraňování nepotřebných bloků trati. Těmi jsou hlavně myšleny krátké zatáčky, z obou stran obklopené rovinkami, jejichž existence v mapě je naprosto zbytečná (velmi krátkou nasnímanou zatáčku je možné projet stejně rychle jako rovinku). Jejich odstranění probíhá tak, že obě rovinky před a za zatáčkou jsou sjednoceny v jediný blok. V mapě tak zůstanou pouze delší zatáčky a rovinky. Krátké zatáčky vyvolané například přejezdem překřížení a drobnými nepřesnostmi při měření jsou smazány. Jakmile je mapa trati zpracována, je uložena do souboru na paměťovou kartu. Hned poté je její zpracovaná i původní podoba vypsána pro kontrolu do logovacího souboru. Pak program přechází do závodního režimu, tedy jízdy dle mapy.





Obrázek 2: Trať ideálně rozdělená na jednotlivé bloky.

### 6.2.2 Kvalita mapování

Kvalitu mapování dráhy ovlivňuje několik faktorů, které je nutné uvažovat při snaze o přesnější nasnímání mapy. Vzhledem k faktu, že velikost přetížení působícího na vůz roste s rychlostí průjezdu tratí, je dobré při stávajícím způsobu zpracování dat zajistit dostatečnou mapovací rychlost a přilnavost na trati. Rychlost ale nesmí být příliš vysoká, aby vůz nevypadl z dráhy. Zvýšení přilnavosti lze dosáhnout odmaštěním pneumatik, případně výměnou pneumatik za nové. Následně naměřená data přesněji vypovídají o profilu závodní dráhy, neboť při rychlejší jízdě jsou patrnější přechody mezi jednotlivými bloky trati.

## 6.3 Závodní režim

Část kódu reprezentovaná převážně funkcí *jizda\_mapou* odpovídá za řízení auta během samotného závodu. A to pouze na základě dat ze snímačů a informací, které byly naměřeny a uloženy během předcházejícího mapovacího cyklu.

Funkce je prvně volána, jsou-li splněny dva základní požadavky. Těmi jsou, jak již bylo zmíněno, úspěšné načtení namapované závodní trati do paměti a projetí přes startovní čáru.



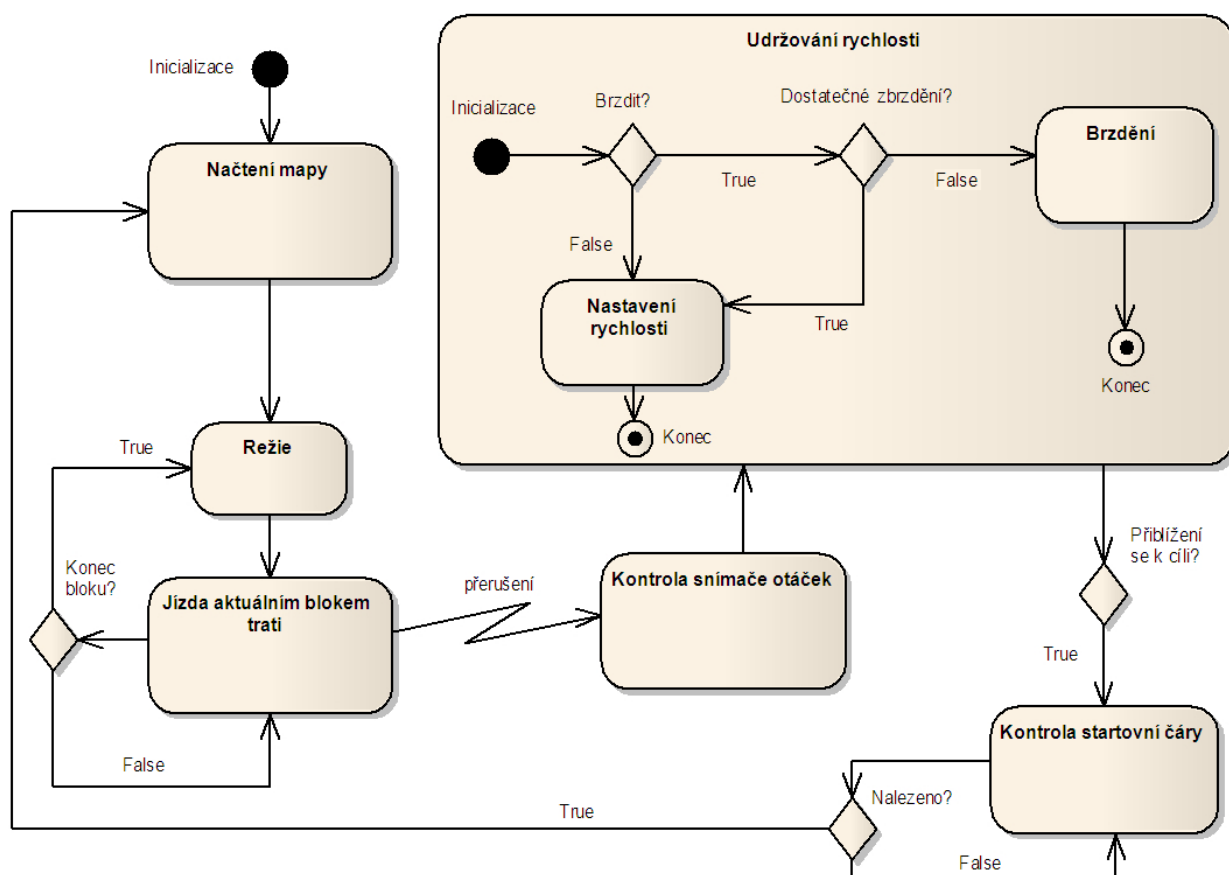
Při přechodu do závodního režimu je nejprve vynulováno interní počítadlo ujeté vzdálenosti. Díky tomu, že se hodnoty vzdálenosti každým průjezdem kola počítají od nuly, je umožněno jejich přímé porovnávání s daty ve vytvořené mapě. Důležité je také nastavit pomocné proměnné, uchovávající v sobě nutné údaje pro řízení jízdy.

Pro potřebu orientace v mapě je třeba nastavit hodnoty dvou proměnných. Tyto proměnné udávají pozici jak aktuálně projížděného, tak po něm následujícího bloku trati. Následně je zahájeno první kolo závodní jízdy.

Celý režim závodní jízdy ve svém principu funguje velmi jednoduše. Jeho hlavní činností je neustále porovnávat hodnotu aktuálně ujeté vzdálenosti s daty, nacházejícími se v mapě trati a pomocných proměnných. Tímto způsobem je možné snadno rozpoznat, ve které části trati (ve kterém bloku) se v každém konkrétním čase vůz nachází, a také co ho následně čeká. Díky tomu je možné projíždět mapou trati od startu až k cíli a patřičně reagovat na měnící se parametry trati.

Během jízdy tratí je kontrolován aktuálně projížděný blok a dle jeho typu je zvolen další postup v programu. Je tak prováděna buď část zabývající se jízdou rovinkou, nebo zatáčkou. Každá z nich je v programu reprezentována obslužným kódem, neustále vykonávaným ve smyčce. Tato část programu je ukončena v momentě, kdy auto dojede na konec aktuálního bloku, tedy když dojede na pozici počátku bloku následujícího.

Tyto zmíněné dvě části jsou v mnoha ohledech rozdílné, ale mají i několik společných prvků. V obou případech je například zjišťováno, zdali se neblíží konec závodního kola. A to díky údajům o délce trati a aktuální ujeté vzdálenosti. Zjistí-li algoritmus, že je auto vzdáleno méně než 40 centimetrů od cíle, aktivuje snímače startovní čáry a čeká na indikaci průjezdu cílem. Nachází se totiž s největší pravděpodobností v cílové rovině nebo těsně před ní a bude tedy nutné zahájit průjezd mapou od začátku. Také je myšleno na případ, kdy by nedošlo k očekávané synchronizaci se startovní čarou. V tom případě vůz zpomalí a pokračuje ve snímání trati do té doby, dokud startovní čáru nenalezne. Tento stav se může vyskytnout v případě poruchy snímačů nebo špatného nasnímání mapy, kdy by délka mapy byla kratší než délka reálné trati. Zjednodušená činnost algoritmu během závodního režimu je patrná viz obr. 3/s. 31.



Obrázek 3: Diagram činnosti algoritmu během jízdy dle naměřených dat.

### 6.3.1 Jízda zatáčkou

Při průjezdu blokem trati typu pravá nebo levá zatáčka zabezpečuje program mimo rutinních úkonů dvě základní činnosti. První z nich je udržování rychlosti jízdy na přijatelné, předem nastavené úrovni tak, aby byla snížena možnost vypadnutí ze závodní dráhy. Jeho druhým úkolem je kontrola pozice. Porovnává hodnotu aktuálně ujeté vzdálenosti s daty v mapě a vyhodnotí-li, že auto dojelo bezpečně na konec zatáčky, posune ukazatel určující aktuální pozici na další blok v mapě. Pokračuje tak jízdou dalším blokem trati.



### 6.3.2 Jízda rovinkou

Poněkud složitější situace nastává v případě jízdy po rovince. Algoritmus musí během jízdy rovinkou rozhodnout, jakým způsobem zrychlit, jak nastavit a udržovat rychlost, a také kdy začít brzdít tak, aby mohl v klidu vjet do následující zatáčky s co nejmenší možností vypadnutí.

V principu vše funguje tak, že jízda každou z rovinek je rozdělena na tři po sobě následující kroky. Nejprve program při vjezdu do rovinky nastaví proměnnou *rychlost* na požadovanou hodnotu *rychlost\_rovinka*. Druhým krokem je jednorázové skokové zvýšení napětí na motoru, které pomůže snazšímu a rychlejšímu dosažení požadované rychlosti průjezdu rovinkou. Na závěr model v případě potřeby brzdí, čímž je dosaženo zpomalení auta před začátkem zatáčky.

Při vjezdu do rovinky je proveden výpočet její délky. Délka bloku je rovna vzdálenosti mezi počátkem aktuálního bloku trati a počátkem následujícího bloku trati. V návaznosti na vypočtenou délku jsou poté stanoveny hodnoty dalších proměnných rozdělení bloku na třetiny.

Moment ve kterém dojde k jednorázovému zrychlení se nachází na konci první třetiny délky bloku a lze jej vypočítat jako  $\text{vzdálenost aktuálního bloku od startu} + \text{délka bloku} / 3$ . Brzdění je v případě potřeby zahájeno vjezdem vozu do poslední třetiny délky bloku, překročením vzdálenosti rovné vztahu  $\text{vzdálenost aktuálního bloku od startu} + 2 * \text{délka bloku} / 3$ . S přihlédnutím k délce bloku poté dojde k případnému upravení vypočtených hodnot tak, aby tyto zajišťovaly bezproblémový průjezd trati.

Všechny zásahy programu do řízení modelu jsou podmíněny kontrolou ujeté vzdálenosti a jejím porovnáním s daty v mapě trati a důležitých pomocných proměnných.

- Urazilo-li auto při jízdě vzdálenost větší než je délka trati, jedná se o chybový stav. Nedošlo k očekávanému nalezení cílové čáry, které by umožnilo přechod do nového kola dráhy. Program se dostal mimo rozsah mapy. Rychlost jízdy se sníží a je pokračováno ve snímání trati. V okamžiku nalezení cíle je opakovaně zahájena jízda mapou.
- Při zahájení jízdy úsekem rovinky je zvýšena hodnota požadované rychlosti jízdy. Patříčná funkce, starající se o obsluhu rychlosti, následně pomalu zvyšuje a udržuje napětí na motoru tak, aby bylo žádané rychlosti dosaženo.



- Je-li aktuální ujetá vzdálenost větší, než vzdálenost pozice skokového zrychlení od startu, ale zároveň stále menší, než vzdálenost místa, ve kterém je třeba brzdít, je jednorázově zvýšeno napětí na motoru a pokračuje se ve zrychlování.
- Posledním možným stavem je ten, kdy auto dojde do místa, v němž je třeba zahájit brzdění. Dojde k nastavení příslušného příznaku brzdění. Následně funkce starající se o udržování rychlosti jízdy začne auto zpomalovat až k žádané hodnotě. Je-li ale rovinka kratší než hodnota proměnné *min\_rovinka*, k aktivnímu brzdění nedochází. Během rovinky nedosáhne vůz takové rychlosti, aby hrozilo vypadnutí z dráhy. Zpomalení je dosaženo prostým snížením napětí na motoru před koncem rovinky.
- Je-li ujetá vzdálenost větší nebo rovna vzdálenosti počátku následujícího bloku trati od startu, znamená to, že auto došlo na konec rovinky. Tudíž dojde k posunu ukazatele aktuálního bloku v mapě trati na další pozici a zahájí se patřičný obslužný kód jízdy zatáčkou.

## 6.4 Řešení vypadnutí z dráhy

Vypadnutí modelu z trati je neblahým momentem, při kterém dojde k resetování procesoru a tudíž k zapomenutí stavu programu, ve kterém se aktuálně nacházel. A to včetně hodnot všech proměnných a nasnímaných parametrů, pokud tyto nebyly předtím uloženy na paměťovou kartu. Auto, znovu nasazené na dráhu po vypadnutí, má za úkol „uvědomit“ si nejprve svou pozici na trati a následně pokračovat v závodní jízdě. Existuje několik způsobů, jakými může auto po vypadnutí z dráhy zjistit svou pozici.

### 6.4.1 Varianta první

První možností by bylo zálohovat data na kartu a spolu s mapou trati ukládat i aktuální pozici v trati, tedy momentální vzdálenost od startu. Při vypadnutí a opětovném nasazení na dráhu by stačilo pouze načíst uložené hodnoty, tím se zorientovat na trati a pokračovat dále v jízdě. To by však platilo pouze v případě nasazení vozu přesně na místo, ve kterém došlo k uložení poslední hodnoty pozice před vypadnutím, což je samozřejmě prakticky nemožné. Nasazení na dráhu v jiném místě, než kde přesně vůz vpadl, by mělo za následek nežádoucí zkreslení a tím pádem zvýšení nepřesnosti v měření ujeté vzdálenosti při další jízdě.



Také je možné po opětovném nasazení na dráhu načíst pozici místa vypadnutí, zjistit si z mapy trati nejbližší následující zatáčku a pomalu se rozjet. Vjezdem do zatáčky program získá údaj o pozici auta (z mapy zná vzdálenost od startu k počátku této zatáčky) a může zahájit závod. Nicméně i tento přístup je problematický. Vzhledem k proměnným jízdním podmínkám není nikdy možné zaručit, že program nasnímá zatáčku ve stejné vzdálenosti od startu jako když při prvním kole, kdy mapoval dráhu. Tím vzniká další nepřesnost v určení polohy, proto není vhodné použít ani tento způsob.

#### 6.4.2 Varianta druhá

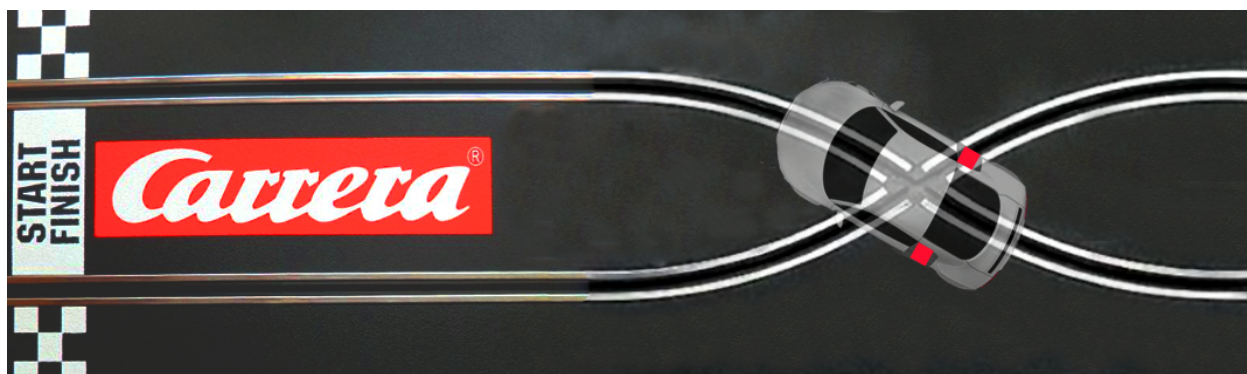
V pravidlech soutěže je řečeno, že pro měření času jízdy bude vždy použita originální časomíra Carrera Evolution. Její polohu lze proto využít jako referenční bod pro měření ujeté vzdálenosti, tedy i pro zjištění pozice v mapě. Časomíra je konstruována jako brána napříč dráhou. Průjezd skrz ni může být zaznamenán, pokud je auto vybaveno patřičnými senzory. Pro tento účel je vhodné využít snímač vzdálenosti, například ultrazvukový senzor, či laserový dálkoměr namířený kolmo ke směru jízdy.

#### 6.4.3 Varianta třetí

Nejjednodušším způsobem pro zjištění aktuální pozice (za účelem synchronizace s mapou trati) je snímání přejezdu překřížení dráhy či průjezdu startovací čarou. Tyto dva způsoby jsou také nejjednodušší na implementaci. Překřížení dráhy je zvláštní díl trati na němž dochází k přejezdu auta z levé dráhy do pravé a naopak. Kolejnice dráhy jsou v bodě překřížení přerušeny. Neustálým kontrolováním napájecího napětí na sběrných kartáčcích lze proto zaznamenat přejezd překřížením jako pokles hodnoty napětí v průběhu delšího časového úseku. Je však nutné zabezpečit, aby program nereagoval na falešně nasnímaná překřížení. Při jízdě po trati se totiž může stát, že dojde k odskočení modelu z dráhy, oddálení sběrných kartáčků a tudíž k chvilkovému výkyvu hodnoty napájení, které samozřejmě nesmí být bráno v úvahu jako relevantní průjezd křížením.

#### 6.4.4 Varianta čtvrtá

Poslední z uvažovaných variant bylo tedy snímání průjezdu startovní čarou. Jak je patrné z předchozích kapitol, byl tento způsob využit při návrhu řídicího algoritmu. Na každé trati



Obrázek 4: Startovní čára, křížení a model s vyznačenými snímači trati.

je použit specifický díl, označující startovní čáru. Je opatřen bílým šachovnicovým polem a bílým pruhem na černém podkladu vozovky viz obr. 4/s. 35. Díky tomuto kontrastnímu prvku je možné pomocí zabudovaných optosnímačů úspěšně indikovat průjezd a získat tak jednoduše údaj o momentální pozici auta. Vypadne-li tedy vůz z trati, po opětovném nasazení na trať pokračuje opatrně v jízdě, přičemž snímá povrch trati. Indikují-li snímače povrchu trati průjezd startovní čarou, je zahájena závodní jízda či mapování trati v závislosti na rozhodnutí programu.

## 6.5 Řízení rychlosti

Během jízdy po dráze je nutné udržovat rychlost na přijatelné úrovni jak v mapovacím kole, tak i při jízdě v závodě. Řízení rychlosti modelu je tudíž důležitou součástí programu. Díky němu je možné proměnlivě nastavovat napětí na motoru a přibližně tak udržovat rychlost jízdy v závislosti na měnících se jízdních podmínkách.

Je-li totiž motor napájen při jízdě pouze konstantní hodnotou napájení, dochází během jízdy v zatáčce vlivem většího tření a vyššího působícího přetížení ke znatelnému zpomalování vozu, oproti rychlosti jízdy na rovině. Zároveň je třeba zajistit zrychlování a zpomalování vozu ve vhodných okamžicích při závodní jízdě. Byla tedy vytvořena funkce *Udrzuj\_rychlost*, jejímž úkolem je právě obsluha rychlosti jízdy tratí.

Funkce je závislá na správné činnosti snímače otáček předního kola. V pravidelných časových intervalech totiž kontroluje množství pulsů ze snímače otáček a následně z jejich počtu rozhodne, jakou rychlostí se vůz pohybuje. Další činnost funkce se dělí na dvě části





podle toho, zda je třeba brzdít či nikoliv. Je-li nutné prudce zpomalit, zahájí funkce aktivní brzdění 6.5.2 a pokračuje v něm až do doby, dokud rychlost auta neklesne pod nastavenou mez. V tom případě je jasné, že vůz dostatečně zpomalil a je možno přejít zpět do režimu udržování rychlosti. Brzdění je důležitým úkonem pro závodní jízdu, kdy je po zrychlení na rovince opět nutné dosáhnout bezpečné rychlosti pro jízdu zatáčkou.

Pokud není třeba brzdít, porovnává funkce vypočtenou rychlost s požadovanými hodnotami a vhodně snižuje, nebo naopak zvyšuje napětí motoru takovým způsobem, aby bylo dosaženo žádané rychlosti jízdy. Zároveň omezuje maximální hodnotu napětí motoru při jízdě v zatáčkách, což přispívá ke snížení pravděpodobnosti vzniku smyku. Kontroluje také, aby napětí na motoru nekleslo pod minimální nastavenou mez.

### 6.5.1 Pasivní brzdění

Pasivní brzdění je nejjednodušší způsob, jak snížit rychlost jízdy auta. Stačí k tomu pouhé snížení napětí na motoru nebo jeho úplné odpojení od napětí, což vlastně představuje situaci, při níž se reálný vůz pohybuje setrvačností po silnici s vyřazeným rychlostním stupněm. Prakticky toho lze dosáhnout deaktivací H-můstku, přivedením nulového signálu na oba dva jeho ovládací vstupy, nebo jeden vstup ponechat nulový a na druhém zmenšit střidu spínacího signálu. Setrvačností tak model pokračuje v jízdě, díky snížené hnací síle motoru však zpomaluje.

Pasivního brzdění je využito při udržování rychlosti, kdy je snižováním napětí na motoru dosahováno drobných korekcí v rychlosti jízdy.

### 6.5.2 Aktivní brzdění

Aktivní brzdění představuje efektivní způsob, jakým snížit rychlost jízdy vozu na závodní dráze. Za pomoci aktivního brzdění lze motorem vynutit větší brzdnou sílu, než jen prostým snížením napětí na motoru. Princip funkce je poměrně prostý. Motoru je nuceně udělena pevná rychlost otáčení, která je za pomoci pneumatik přenášena na trať. Pevně určená rychlost otáčení kol tak působí brzdnou silou a zpomaluje model do té doby, než se rychlost auta ustálí na rychlosti dané otáčkami kol. V praxi se tento druh zpomalování dá přirovnat k brzdění reálného auta motorem za pomoci zařazeného rychlostního stupně. Například při jízdě z kopce.





Přivedením totožného nenulového signálu na oba ovládací vstupy lze také dosáhnout, v závislosti na rychlosti jízdy, okamžitého zablokování kol hnané nápravy. Čím větší je rychlost jízdy autíčka, tím větší úroveň ovládacích signálů potřebujeme pro udržení zablokováných, či pevnou rychlostí se otáčejících kol. Nízká úroveň signálů totiž nemá dostatečnou sílu na udržení přesných otáček, nebo zastavených kol a rychlost jízdy auta by ji snadno přemohla.

Blokace kol však není využitelná jako nástroj pro efektivní brzdění. Zablockovaná kola totiž neposkytují dostatečnou brzdnu sílu. Nejlepšího brzdění je dosaženo koly, jenž se stále otáčejí. Zablockovaná kola také způsobují smyk zadní nápravy a zhoršují chování vozu během brzdění.

Aktivní brzdění je tedy vyvoláno přivedením PWM signálů na oba ovládací vstupy H-můstku. Vzájemný poměr středních hodnot signálů následně určuje chování motoru. Pro brzdění musí mít signál určující dopředné otáčky vyšší střední hodnotu, než je střední hodnota signálu pro zpětný chod. Toho je dosaženo volbou střídý signálů ve vhodném poměru nastavením registrů *TPM2C0V* a *TPM2C1V* v rozsahu 0–6000, přičemž hodnota nula nastaví maximální možnou hodnotu výstupního signálu k motoru. První ovládací PWM signál je procesorem generován pravidelně a umožňuje jízdu vozu vpřed. Generování druhého signálu je aktivováno nastavením registru *TPM2C0SC*.<sup>[4]</sup>

Při malých rychlostech je rozdíl v brzdné dráze mezi aktivním a pasivním brzděním nepatrný, při zpomalování z vyšších rychlostí je už rozdíl o mnoho výraznější, což bylo zjištěno pozorováním chování vozu na dráze.

## 6.6 Ukládání dat na paměťovou kartu

Díky dostatečné velikosti paměti RAM je možné mikrokontrolér využít pro přístup k paměťové kartě. Hardwarově je karta připojena k procesoru pouze několika vodiči. Těmi jsou *MOSI*<sup>9</sup>, *MISO*<sup>10</sup>, *CLK*<sup>11</sup>, *CS*<sup>12</sup> a samozřejmě je nutné přivést ke kartě napájení. Pin *MOSI* je vstup obstarávající proudění dat z mikroprocesoru na kartu, *MISO* poskytuje data v opač-

---

<sup>9</sup>Master Out, Slave In

<sup>10</sup>Master In, Slave Out

<sup>11</sup>Clock

<sup>12</sup>Chip Select



ném směru. *CLK* je vstup pro taktovací signál, nepostradatelný pro chod sériové linky a *CS* je aktivační signál určující, zda karta reaguje na data proudící skrze rozhraní SPI, či nikoliv. Více o komunikaci s paměťovou kartou lze nalézt v literatuře viz [8].

Softwarově přistupuje mikroprocesor k paměťové kartě skrze souborový formát Fat FS, poskytující základní funkce pro práci jak s paměťovým prostorem, tak i s vlastními soubory.

Ukládání dat na paměťovou kartu je velmi důležité a je využito pro tvorbu logovacích souborů, do nichž jsou v pravidelných intervalech ukládány informace o průběhu jízdy. Jejich rozbořem pak lze zpětně kontrolovat správnou funkci programu.

Paměťové karty je v programu také využito pro uložení souboru s nasnímanou mapou trati, kterou je výhodné uložit ihned po nasnímání. V případě nutnosti je tak možné mapu kdykoliv načíst a použít.

Důvodem proč využívat ukládání dat na paměťovou kartu, přestože je možné data uchovávat za běhu programu v paměti mikrokontroléru je fakt, že veškerá nasnímaná data, která ještě nebyla uložena na paměťové medium, mohou být velmi snadno ztracena. V případě že nastane výpadek napájení, dojde ke ztrátě dat v paměti mikrokontroléru. Krátkodobé výpadky v napájení lze úspěšně pokrýt připojením kondenzátoru na vstupní svorky napájení řídicí elektroniky.

Doba, po kterou je kondenzátor schopen napájet plošný spoj, je závislá také na přidaném hardwaru a jeho zapojení. Jelikož jsou jak snímač otáček, tak oba snímače povrchu startovní dráhy napájeny z obvodu stejně jako mikrokontrolér, odebírají i ony neustále z obvodu energii. Na pokrytí delších výpadků napájení způsobených vypadnutím z dráhy tudíž není připojený kondenzátor dostatečný.

Prodloužení doby, po kterou je kondenzátor schopen napájet obvod, je možné dosáhnout optimalizací napěťových nároků přídatného hardware a použitím vysokokapacitního kondenzátoru. Cesta zmenšení napěťových nároků s sebou také nese možnost použití malých nízkopříkonových integrovaných obvodů a čidel, které současně vyžadují méně prostoru pro zabudování do stísněné karoserie auta. Tím by bylo zároveň dosaženo i snížení celkové hmotnosti vozu.

**Fat FS** je volně dostupný souborový systém určený pro malé vestavné systémy. Jeho prostřednictvím je přistupováno k paměťové kartě. Je nezávislý na hardwarové platformě a lze



ho využívat s celou řadou mikrokontrolérů. Umožňuje na datovém nosiči vytvářet a mazat soubory, číst z nich data a zapisovat do nich. Z dostupné knihovny souborového systému je v programu využito jen několik základních funkcí. Kontrolou návratové hodnoty každé volané funkce lze vyhodnotit úspěšnost jejího provedení, případně příčinu chyby.[1]

- **f\_\_mount** – Zajišťuje připojení paměťového media.
- **f\_\_open** – Vytvoření objektu, umožňujícího práci se souborem.
- **f\_\_write** – Umožňuje zápis dat do souboru.
- **f\_\_read** – Slouží pro čtení dat ze souboru.
- **f\_\_printf** – Zápis formátovaného textu do souboru.
- **f\_\_sync** – Fyzické uložení dat na kartu.
- **f\_\_close** – Uložení dat a uzavření souboru.



## 7 Nadstandardní HW výbava

Během vývoje bylo rozhodnuto o potřebě využití dalších hardwarových prvků k tomu, aby bylo možné vytvořit jednoduchý program, který by vůz řídil. Osazení dodatečného hardwaru do auta nepředstavuje dle pravidel žádný problém, je pouze třeba se držet několika pokynů pro jeho zástavbu. Nesmí být překročena maximální přípustná hmotnost auta a žádný ze snímačů nesmí přesahovat vnější rozměry karoserie. Při volbě snímačů bylo čerpáno z literatury viz [12].

Uvnitř vozu je tedy vložen dodatečný plošný spoj, umožňující připojení signálů až od čtyř snímačů CNY70. Jedná se o reflexní optosnímače, jenž obsahují ve společném pouzdře jak zdroj světelného záření, tak i fototranzistor. Jako emitor záření je použita infradioda s vlnovou délkou světla 950 nm.[13]

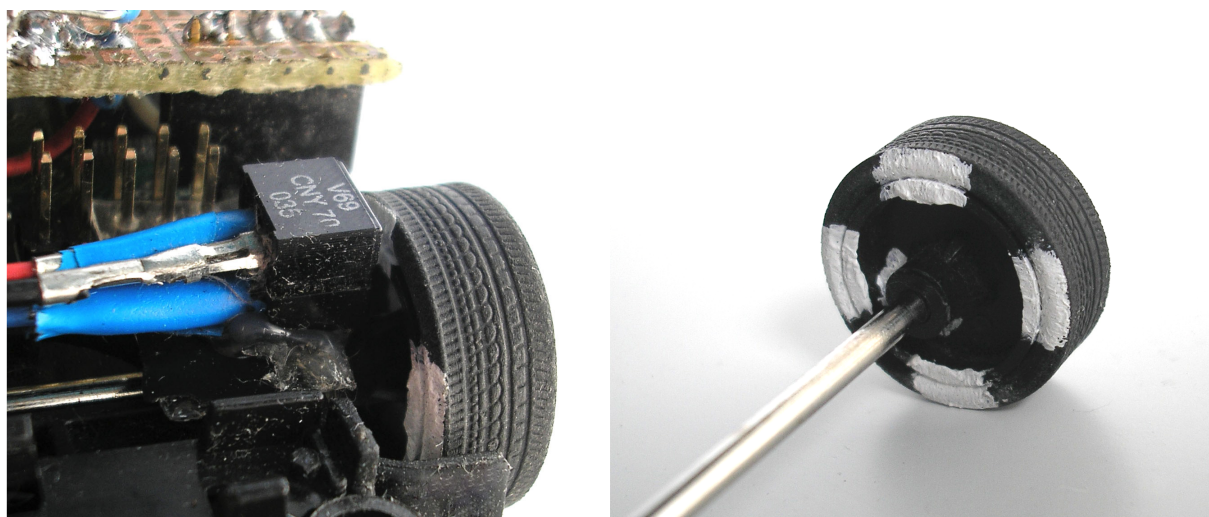
Umístění obou dvou prvků paralelně vůči sobě umožňuje fototranzistoru zachytávat a zesilovat emitované záření odražené od snímané plochy před senzorem. Vhodným připojením snímače do obvodu a jeho kolmým umístěním vůči snímané ploše lze na výstupu fototranzistoru získávat signál, jehož amplituda je přímo úměrná množství zachyceného odraženého světla. Další informace o obvodu CNY70 je možno získat v literatuře [13].

Výstupní signály z použitých snímačů je nejprve potřeba upravit viz 7.3 a poté přivést na čtveřici pinů procesoru. Tyto piny jsou značeny *CON10* a *CON11* a lze je najít ve schématu viz obr. 12/s. 54.

### 7.1 Snímač ujeté vzdálenosti

Nutností se ukázalo zajištění určování pozice auta na závodní trati. Vytvoření snímače otáček, tedy snímače ujeté vzdálenosti, bylo důležitou základní podmínkou dalšího postupu v práci. Nejprve bylo nutné zhodnotit, jakým způsobem nejjednodušeji zjišťovat ujetou vzdálenost.

Jako první možnost se jevilo použití enkodéru z počítačové myši, či vyřazené tiskárny, což je vlastně soustava složená z fototranzistoru, světlo emitující diody a děrovaného kolečka mezi nimi. Otáčením kolečka se postupně zakrývá a odkrývá fotocitlivá plocha fototranzistoru a na jeho výstupu tudíž dochází ke generování pulsů. Napočítáme-li na výstupu tranzistoru počet pulsů rovný počtu otvorů v kolečku enkodéru, pak víme, že se kolečko otočilo kolem své osy právě jednou. Výhodou tohoto řešení by bylo velmi přesné měření odvíjející



Obrázek 5: Snímač otáček a reflexní značky na kole.

se od počtu otvorů v kolečku enkodéru a snadná realizace, protože se jedná o již přesně sestavenou a odzkoušenou součástku. Na druhou stranu problém by představovala nutnost nasadit kolečko enkodéru na hřídelku snímané nápravy. Vzhledem k možnému riziku poškození kolečka autíčka při jeho demontáži a prostorové náročnosti zabudování tohoto snímače do stísněného prostoru v karoserii bylo použití tohoto snímače shledáno jako nevhodné. Velkou měrou k tomuto rozhodnutí přispěla i snaha veškerý přidaný hardware koncipovat jako vyjímatelný a snadno zaměnitelný, či nahraditelný jiným v případě budoucího navázání na tuto práci.

Druhým způsobem, jak jednoduše měřit ujetou vzdálenost autíčka, je použití reflexního optosnímače CNY70, jehož vlastností je možné snadno využít pro počítání otáček kol autíčka. Na černé zadní straně kola auta byly vytvořeny čtyři bílé značky, pravidelně rozmístěné po celém jeho obvodu viz obr. 5/s. 41. Tímto došlo k získání ploch o různé odrazivosti záření a vznikl tedy vhodný objekt pro snímání senzorem. Bílá barva odráží větší množství světla než černá. Proto se každý průchod značky v prostoru před snímačem projeví jako změna amplitudy spojitého signálu na výstupu fototranzistoru. Podle počtu nasnímaných odrazů lze získat hodnotu ujeté vzdálenosti. Známe totiž obvod kola a fakt, že každá změna signálu reprezentuje otočení kola o čtvrt otáčky.



Pro zajištění funkčnosti optosnímače je třeba zabezpečit jeho správné zapojení a umístění dovnitř karoserie ve vzdálenosti 2–3 mm od zadní strany kolečka. Také je třeba kontrolovat celistvost bílých značek a v případě potřeby je vhodnou barvou obnovit.

### 7.1.1 Obsluha měření ujeté vzdálenosti

Signál ze snímače je po tvarování přiveden na vstup GPIO mikroprocesoru. Hodnota na pinu je programově kontrolována s frekvencí 2 kHz funkcí *kontrola\_GPIO*. Její princip je jednoduchý. Signál je aktivní v nule, proto je-li hodnota pinu rovna logické nule (nedochází k odrazu, před snímačem je černá), právě jednou inkrementuje vnitřní počítadlo pulsů, pomocí něhož je následně počítána ujetá vzdálenost. Ta je rovna počtu pulsů vynásobených 1,66 cm, což je právě čtvrtina obvodu kolečka.

Jelikož je doba přechodu bílé značky na pneumatice autíčka před snímačem několikanásobně delší, než pravidelný interval se kterým je volána funkce pro kontrolu snímače ujeté vzdálenosti, je třeba zabezpečit, aby funkce zvýšila vnitřní počítadlo pouze jednou během průchodu jedné a té samé značky. Proto nasnímá-li optosnímač černou barvu, pozastaví další zvyšování počítadla ujeté vzdálenosti do té doby, dokud není před snímačem opět bílá barva (černá plocha minula snímač).

### 7.1.2 Přesnost měření

Na přesnost měření ujeté vzdálenosti má vliv hned několik faktorů. Například počáteční poloha autíčka po jeho položení na dráhu. Kdy po spuštění programu není značka před senzorem v ideální pozici a po rozjezdu dojde k jejímu nasnímání dříve, než by bylo správné. Dojde tak k nepřesné indikaci otočení o čtvrt otáčky. Tato chyba by se dala zmenšit zvýšením počtu reflexních bodů. Vzhledem k velikosti snímače a širokému kuželu emitovaného světla, tedy parametrům, jenž určují minimální velikost reflexní značky je toto nemožné. Přesto však není případná chyba velká a neznamena tak velký problém.

Je také nutné dbát na správný kontakt kol přední nápravy s povrchem trati. Oddálí-li se kola od povrchu, zastaví se na čas jejich otáčení a měření ujeté vzdálenosti tím pádem snímá data s chybou. Kontakt s vozovkou může být také negativně ovlivňován nevhodně tvarovanými a opotřeбенými sběrnými kartáčky, nečistotami na trati, opotřeбенými pneumatikami



a celkovou hmotností auta. Vyšší hmotnost vozu totiž přispívá k lepšímu kontaktu přední nápravy z povrchem trati.

Další chyba měření je způsobena absencí diferenciálu na přední nápravě. V zatáčkách se tudíž kola na společné hřídeli do jisté míry vzájemně ovlivňují. Tím pádem výsledná ujetá vzdálenost závisí na tom, které z kol v zatáčce udávalo otáčky nápravy. Během testování se ukázalo, že tato chyba měření se do jisté míry opakuje pravidelně jak při mapování, tak při závodní jízdě. Tudíž jsou naměřená data v jednotlivých kolech sice mírně nepřesná, nicméně tato nepřesnost se opakuje na stejných místech s přibližně stejnou hodnotou.

## 7.2 Snímače startovní čáry

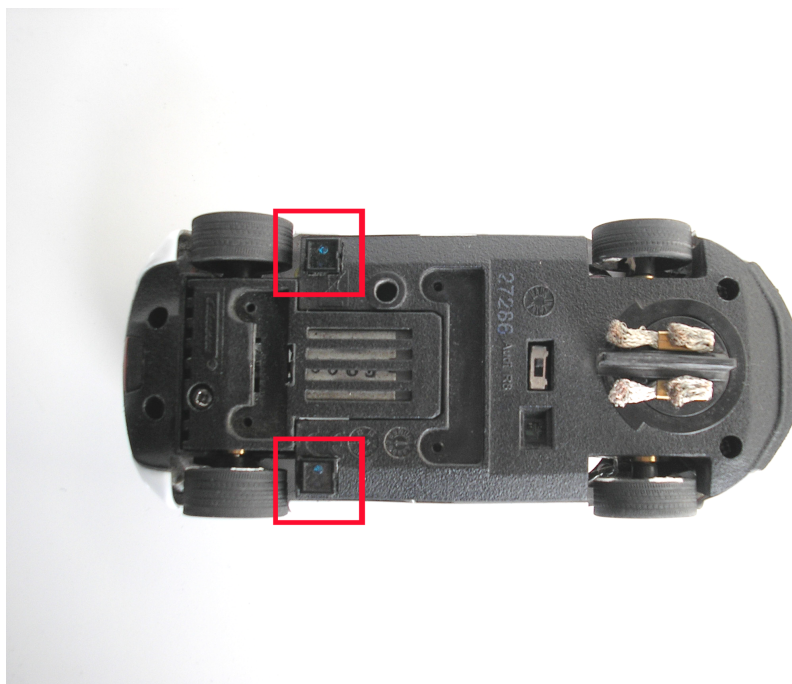
Jedním ze základních předpokladů pro správnou funkci programu je schopnost správně rozpoznat dokončení kola trati. Jedno kolo trati je chápáno jako úsek mezi dvěma průjezdy startovní čarou. Bylo tudíž nutné zvolit vhodný způsob jakým indikovat, kdy tyto momenty nastávají.

Startovní díl je specifická část trati s konektory pro připojení kolejnic ke zdroji napájení. Také je na jeho povrchu vyznačen bílý pruh a souběžné bílo-černé šachovnicové pole. Matná plocha dráhy neodráží dostatečné množství záření, zatímco bílé prvky ano. Díky čemuž je možné použít optoreflexní snímače.

Startovní čára a šachovnice nejsou však jedinými reflexními body dráhy, při jejichž přejezdu dochází k pozitivnímu odrazu. Kvalitní odraz totiž poskytují i napajecí kolejnice trati. Snímaný prostor pod vozem je po většinu času tvořen pouze samotnou plochou trati, vždy při přejezdu překřížení a nebo v případě smyku dojde k nasnímání kolejnic. Z toho důvodu nepostačuje pro snímání startovní čáry použití jen jednoho snímače. Pouze s jeho pomocí by totiž nebylo možné jednoduše rozhodnout o původu odrazu.

Senzory CNY70, tentokrát tedy v páru, byly umístěny před zadní nápravou. Za účelem jejich zabudování bylo třeba vybrousit dva čtvercové otvory v podvozku vozu, do nichž jsou snímače bez nutnosti lepení nasunuty viz obr. 5/s. 41. Snímače je nutné umístit v podvozku symetricky tak, aby se nacházely v jedné rovině kolmé na středovou osu vozu. A to z toho důvodu, aby oba snímače sledovaly stejnou část trati pod autem. Pro jejich správnou funkci je třeba zajistit vzdálenosti mezi snímači a vozovkou zhruba 2–3 mm.





Obrázek 6: Snímače startovní čáry a jejich umístění.

### 7.2.1 Nepřesnost snímání

Snížení počtu falešně indikovaných odrazů je dosaženo tím, že jako relevantní je odraz uvažován pouze v případě, zahlásí-li ho během několika po sobě jdoucích snímáních obě čidla zároveň. Díky tomu lze úspěšně odfiltrout odrazy vyvolané přejezdem překřížení. V tomto případě totiž nikdy nenastane odraz na obou snímačích zároveň. Průjezd auta překřížením je patrný viz obr. 4/s. 35.

Problematický moment však stále nastává v případě, kdy se auto dostane do smyku. Vzdálenost mezi senzory bohužel umožňuje, aby auto, vybočující zadní částí do prostoru vedlejší trati, nasnívalo odraz od obou napájecích kolejnic zároveň. Došlo by tak k chybnému zaznamenání cíle. Smyk během mapování zpravidla nenastává, zatímco v režimu závodní jízdy je to možné. Proto jsou během závodního kola data ze senzorů ignorována a brána v potaz jsou pouze za předpokladu, že je auto ve vzdálenosti menší, nebo rovné 40 cm před cílem.





### 7.2.2 Obsluha snímání

O kontrolu dat ze snímačů se stará funkce *snimani\_startovni\_cary*. V případě, že je povoleno kontrolovat trať, je funkce pravidelně volána s frekvencí 2000 Hz. Jakmile obě čidla nasnímají odraz, inkrementuje se pomocná proměnná. Dosáhne-li hodnota proměnné požadované velikosti, čeká funkce na okamžik, kdy k odrazu přestane docházet. Následně funkce indikuje start tím, že přičte jedničku k proměnné počítající zaznamenané průjezdy. Průjezd cílem je tedy zaznamenán těsně po přjetí startovní čáry, kdy snímače opustí prostor nad bílým pruhem a šachovnicí a objeví se znovu nad tmavou vozovkou.

Každý průjezd startem vynuluje ujetou vzdálenost a podle počtu projetí startu funkce rozhoduje o další činnosti programu. První průjezd zahájí mapování trati. Další průjezd ukončí mapování, a zahájí závod. Všechny další průjezdy znamenají zastavení snímání trati a zahájení jízdy dle uložených dat.

Existuje-li při spuštění programu mapa trati na paměťové kartě, je počítadlo průjezdů nastaveno na vyšší hodnotu než dvě. Díky tomu je fyzicky první průjezd považován již za několikátý a je tak okamžitě zahájen závod.

## 7.3 Tvarování signálu snímačů

Optosnímače generují spojitý signál, jehož hodnota je úměrná množství odraženého světla mezi infradiodou a fototranzistorem. Proto chceme-li získat přesnou aktuální hodnotu signálu, je třeba na patřičných vstupech procesoru konfigurovat analogově-digitální převodník a s jeho pomocí signál vzorkovat. Senzory jsou však použity pouze pro indikaci, nikoliv pro měření přesné hodnoty. Díky tomu stačí, aby program pouze obdržel digitální hodnoty nabývající logické jedničky či nuly. Tedy zdali k odrazu dochází či nikoliv. Z toho důvodu bylo nutné výstupní signál všech čidel patřičně upravit.

Mikroprocesor je napájen konstantním napětím o hodnotě 3,3 V, stejným napětím jsou napájeny i snímače a tvarovací obvod. Proto byl pro úpravu signálu zvolen integrovaný obvod 74HC132 postavený na bázi logiky CMOS<sup>13</sup>, který dokáže pracovat s napájecím napětím v rozsahu 2–6 V[11].

---

<sup>13</sup>Complementary metal-oxide-semiconductor



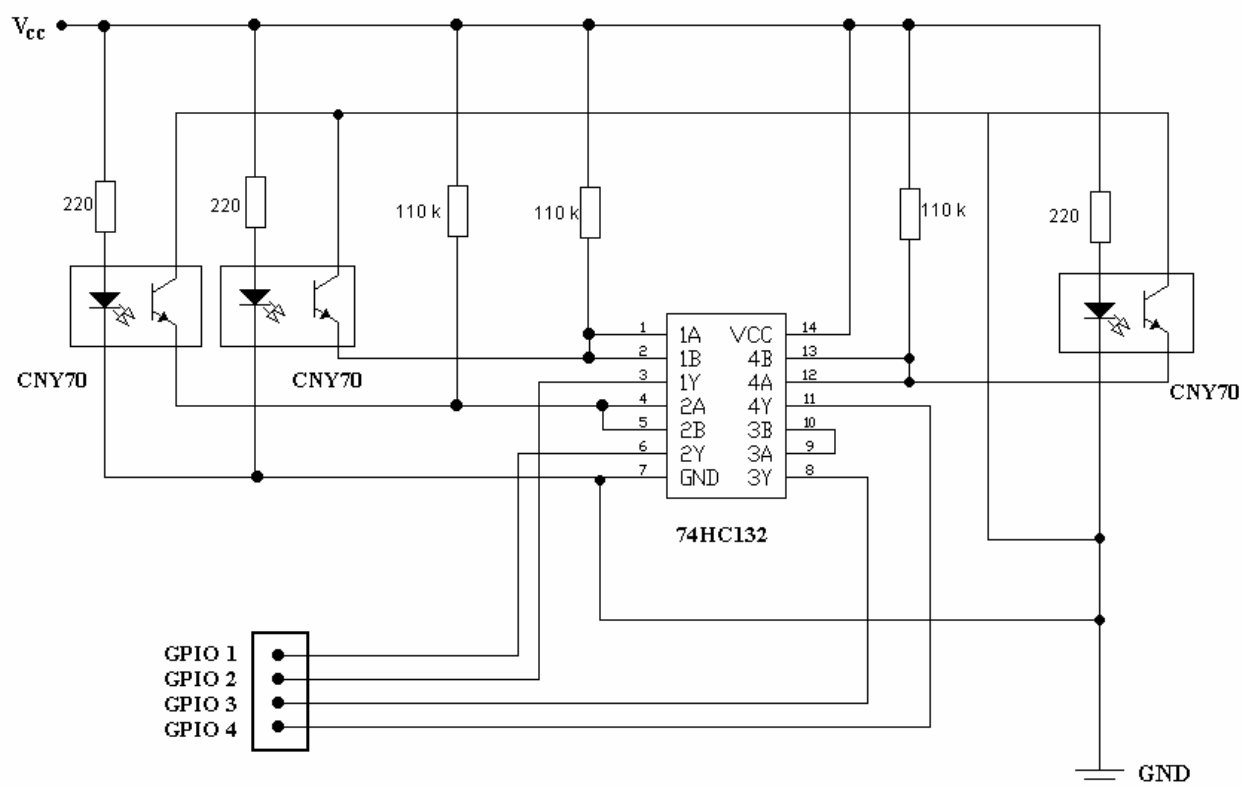
Ve společném pouzdře integrovaný obvod obsahuje čtyři dvouvstupová Schmittova hradla typu NAND. Hradla s hysterezí jsou schopna transformovat vstupní, pomalu se měnící signál, na skokově se měnící výstupní signál bez zvlnění. Funkce obvodu je poměrně jednoduchá. Je-li na vstup hradla přiveden signál o napěťové úrovni menší, nežli  $U_{t+}$ , pak hradlo na výstupu generuje logickou nulu. Vzroste-li napětí na vstupu nad hodnotu  $U_{t+}$ , je výstupní signál obvodu roven logické jedničky. Tato hodnota zůstane na výstupu obvodu do té doby, dokud vstupní napětí neklesne pod hodnotu  $U_{t-}$ . Rozdíl mezi napětími  $U_{t+}$  a  $U_{t-}$  se nazývá hystereze. Bližší údaje o tomto obvodu lze nalézt v literatuře [11].

Dodatečný plošný spoj ve voze plní tedy funkci připojení snímačů k obvodu, tvarování signálu a jeho postoupení dále na piny procesoru. Zapojení použité na závodě bylo vytvořeno s integrovaným obvodem v pouzdře *DIP14*. Schéma obvodu pro úpravu signálu snímačů je zobrazeno viz obr. 7/s. 47.

Jednotlivá hradla obvodu jsou použita jako invertory, tedy oba jejich vstupy jsou spojeny dohromady. Ke každému vstupu hradla je připojen pull-up odpor a signál ze snímače. Pokud je před snímačem černá plocha, na vstupu hradla je napětí 2,4 V zajištěné pull-up odporem. Výstupem hradla je v tuto chvíli logická nula. Naopak odráží-li bílá plocha světlo, fototranzistor spojí vstup hradla se zemí a napětí tudíž klesne na hodnotu 1,2 V. Výstup hradla následně skokově přejde na úroveň logické jedničky.

## 8 Problémy

V průběhu práce na zadaném tématu nastalo několik problematických momentů, které bylo nutné řešit. Největší závadou bylo zničení samotného mikroprocesoru. Během jízdy došlo k uvolnění vodiče traťového napájení, který se zaklínil o vstupní piny procesoru. Bylo tedy nutné procesor nahradit novým kusem a po záměně vyzkoušet funkčnost obvodu jako celku. Správná funkce žádné z ostatních součástí obvodu nebyla zničením procesoru ovlivněna. Nebylo proto nutné provádět žádné další opravy.



Obrázek 7: Zapojení tvarovacího obvodu při použití tří snímačů.

## 8.1 Adheze

Přilnavost autíčka na trati se může během času dost měnit. Je proto třeba přizpůsobit vhodně proměnné programu vždy tak, aby byla zajištěna kvalitní jízda vozu. Adhezi negativně ovlivňují nečistoty na dráze, její mastnota a také stav pneumatik a vodicího dílu autíčka.

## 8.2 Akcelerometr

Akcelerometr jako takový se zdá být velmi robustní a spolehlivou součástí, přesto se však několikrát ukázal jako problematický. Během příprav na soutěž, které se mělo auto zúčastnit, se několikrát objevila poměrně závažná chyba. Velmi znepríjemňovala programování, testování a vůbec řízení modelu. Akcelerometr se v některých momentech stal méně citlivým



a docházelo k chybám při průjezdu levou zatáčkou. Obvod následně zatáčku indikoval pouze při velkém působícím přetížení, což znamená zhruba v půli průběhu zatáčky, nebo zatáčku neindikoval vůbec. Jak je patrné z funkce akcelerometru a způsobu načítání dat z jeho výstupů, bylo možné tento nahodile se vyskytující problém řešit programově. A to snížením prahových hodnot, při jejichž překročení mikroprocesor indikuje zatáčku. Tento programový zásah ale v důsledku způsobil větší množství indikovaných zatáček na úkor nasnímaných rovinek, díky čemuž model neměl mnoho prostoru pro zrychlování.

Tato chyba se náhodně vyskytovala během intenzivního testování před soutěží. Přehrání aktuálního řídicího programu do jiného modelu prokázala, že chyba snímání musí být na straně akcelerometru a ne programu, který v druhém modelu pracoval dle předpokladů. Po delší pauze v dalších fázích práce se již tento problém znovu nevyskytoval.

### 8.3 Nastavení před jízdou

Před jízdou je nutné zkontrolovat stav několika důležitých součástí proto, aby byla zajištěna správná funkčnost programu. Předně, celý program je závislý na funkčnosti snímače ujeté vzdálenosti. Pokud by nefungoval jak má, modelu by se nedařilo udržovat rychlost jízdy. V případě poruchy snímače by program vyhodnotil, že jede vůz příliš pomalu a snažil by se tedy neustále dosáhnout žádané rychlosti zvyšováním napětí na motoru. Proto by následně došlo k vypadnutí z dráhy vlivem vysoké rychlosti. Správná funkce snímače ujeté vzdálenosti je podmíněna celistvostí bílých reflexních značek snímaného kola, stálým kontaktem přední nápravy s vozovkou a stabilní pozici snímače vůči kolu.

Žádanou rychlost jízdy je třeba nastavit také s ohledem na stav pneumatik. Samozřejmě je použití kvalitních nesjetých pneumatik, které vůz drží lépe na vozovce. Nedochází pak tak často ke smyku a prokluzu kol. Jsou-li pneumatiky sjeté neustálým testováním, snadno se mastí a nachytává se na ně nejrůznější špína. Je proto nutné je občas odmastit a očistit.



## Závěr

V rámci práce byl vyvinut jednoduchý řídicí algoritmus, který ve spolupráci s navrženými snímači povrchu dráhy, snímačem ujeté vzdálenosti a s daty naměřenými akcelerometrem umožňuje autu jízdu po závodní dráze. S tímto algoritmem bylo možno reprezentovat Technickou univerzitu v Liberci na univerzitním finále soutěže Freescale Race Challenge, jenž se konalo v areálu Českého vysokého učení technického v Praze dne 30. března 2011. Do soutěže bylo přihlášeno 34 závodních týmů z nichž 21 se závodů skutečně zúčastnilo. V hlavním závodě na neznámé trati se auto umístilo v celkovém součtu na 18. místě. Ve vyřazovacím turnaji se autu vedlo o poznání lépe a proboujvalo se mezi osm nejrychlejších vozů bojujících o postup do semifinále.

Na soutěži bylo k vidění několik zajímavých modelů s velmi kvalitními řídicími algoritmy, kterými by se dalo inspirovat při dalším vývoji. Některé z modelů disponovaly velmi propracovaným způsobem zpracovávání dat z akcelerometru, díky kterému byly schopné přesně řídit způsob jízdy vozu na závodní trati. Jiné modely zase využívaly ke své funkci mnoho zajímavých hardwarových komponent. Lze zmínit použití zdroje, napájecího model při vypadnutí z dráhy, obvod pro skenování povrchu závodní trati za účelem snímání odlišného zakřivení napájecích kolejnic v zatáčkách a na rovinkách, či ultrazvukového dálkoměru.

Vývoj řídicího algoritmu nebyl místy snadný. I ty nejmenší změny v kódu programu bylo nutné okamžitě testovat na závodní dráze v laboratoři, sledovat jejich vliv na chování vozu při jízdě a následně kontrolovat zápis o činnosti programu na paměťové kartě. Měnící se adheze vozu mezi jednotlivými jízdami a občasné nepřesnosti ve snímání otáček kol přední nápravy však viditelně ovlivňují chování vozu na závodní dráze. Proto bylo místy složité odhadnout, zdali se změna v kódu programu projevila žádaným způsobem během jízdy.

Program byl v průběhu práce několikrát přepracován, aby v sobě kombinoval jednoduchost a přehlednost při zachování požadované funkcionality. Úkolem této práce bylo také nastínit jednu z cest, kterou se lze při tvorbě řídicího algoritmu vydat.

Do budoucnosti je tudíž možné na tuto práci navázat, případně se jí inspirovat a nalézt tak vlastní způsob řešení daného problému. Použité snímače se v praxi osvědčily a lze jich využít při řešení podobných projektů. Při dalším vývoji je možné využít vytvořeného algoritmu jako výchozího bodu pro tvorbu dalších řídicích algoritmů fungujících na obdobném principu.



## Literatura

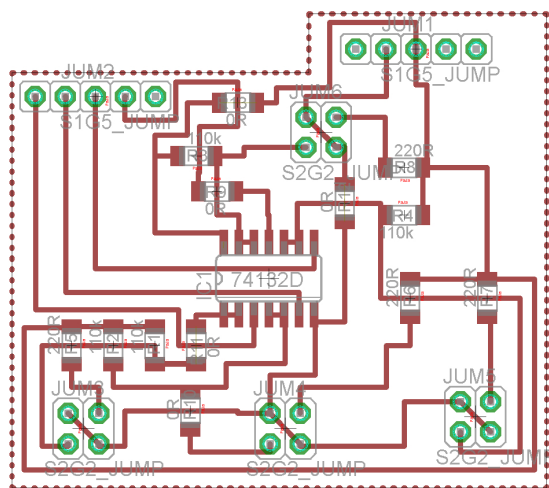
- [1] ChaN: *ELM – FatFs Generic FAT File System Module*. [online], 2010, [cit. 2011-12-18]. URL: [http://elm-chan.org/fsw/ff/00index\\_e.html](http://elm-chan.org/fsw/ff/00index_e.html)
- [2] Freescale Semiconductor, Inc.: *5.0 A Throttle Control H-bridge MC33931*. [online], 2008, [cit. 2011-12-18]. URL: [http://www.freescale.com/files/analog/doc/data\\_sheet/MC33931.pdf](http://www.freescale.com/files/analog/doc/data_sheet/MC33931.pdf)
- [3] Freescale Semiconductor, Inc.:  *$\pm 1.5g$ ,  $\pm 6g$  Three Axis Low-g Micromechined Accelerometer MMA7361L*. [online], 2008, [cit. 2011-12-18]. URL: [http://www.freescale.com/files/sensors/doc/data\\_sheet/MMA7361L.pdf](http://www.freescale.com/files/sensors/doc/data_sheet/MMA7361L.pdf)
- [4] Freescale Semiconductor, Inc.: *MCF51JM128 ColdFire® Integrated Microcontroller Reference Manual*. [online], 2009, [cit. 2011-12-18]. URL: [http://www.freescale.com/files/32bit/doc/ref\\_manual/MCF51JM128RM.pdf](http://www.freescale.com/files/32bit/doc/ref_manual/MCF51JM128RM.pdf)
- [5] Freescale Semiconductor, Inc.: *FRC2011-SlotCarHW*. [online], 2010, [cit. 2011-12-18]. URL: [www.hw.cz/files/FRC2011-SlotCarHW.zip](http://www.hw.cz/files/FRC2011-SlotCarHW.zip)
- [6] Freescale Semiconductor, Rožnov pod Radhoštěm: *Freescale Race Challenge 2011*. [online], 2010, [cit. 2011-12-18]. URL: <http://www.hw.cz/files/FRC2011-IntroductionCZ.pdf>
- [7] Husák, M.: *Mikrosenzory a mikroaktuátory*. Praha : Academia, první vydání, 2008, ISBN 978-80-200-1478-8, 248 s.
- [8] KINGMAX Semiconductor, Inc: *MicroSD Card Specification*. [online], 2007, [cit. 2011-12-18]. URL: <http://www.kingmax.com/material/download/3/microsd.pdf>
- [9] Novák, P.: *Mobilní roboty – pohony, senzory, řízení*. Praha : Nakladatelství BEN, první vydání, 2005, ISBN 80-7300-141-1, 248 s.
- [10] ON Semiconductor: *100mA, Low Power Low Dropout Voltage Regulator LP2950-D*. [online], 2010, [cit. 2011-12-18]. URL: [http://www.onsemi.com/pub\\_link/Collateral/LP2950-D.PDF](http://www.onsemi.com/pub_link/Collateral/LP2950-D.PDF)



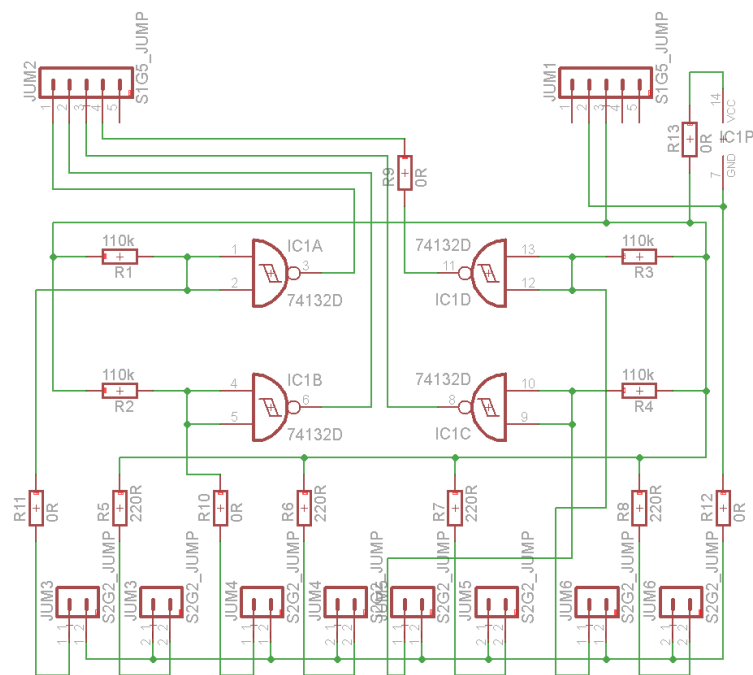
- [11] ON Semiconductors: *MC74HC132A – Quad 2-Input NAND Gate with Schmitt-Trigger Input*. [online], 2011, [cit. 2011-12-18]. URL: [http://www.onsemi.com/pub\\_link/Collateral/MC74HC132A-D.PDF](http://www.onsemi.com/pub_link/Collateral/MC74HC132A-D.PDF)
- [12] Ďádo S.; Kreidl M.: *Senzory a měřicí obvody*. Praha : ČVUT, druhé vydání vydání, 1999, ISBN 80-01-02057-6, 315 s.
- [13] Vishay Semiconductors: *Reflective Optical Sensor with Transistor Output CNY70*. [online], 2009, [cit. 2011-12-18]. URL: <http://www.vishay.com/docs/83751/cny70.pdf>

## A Přílohy

### A.1 Návrh desky pro připojení snímačů



Obrázek 8: Návrh SMD desky pro připojení čtveřice snímačů.

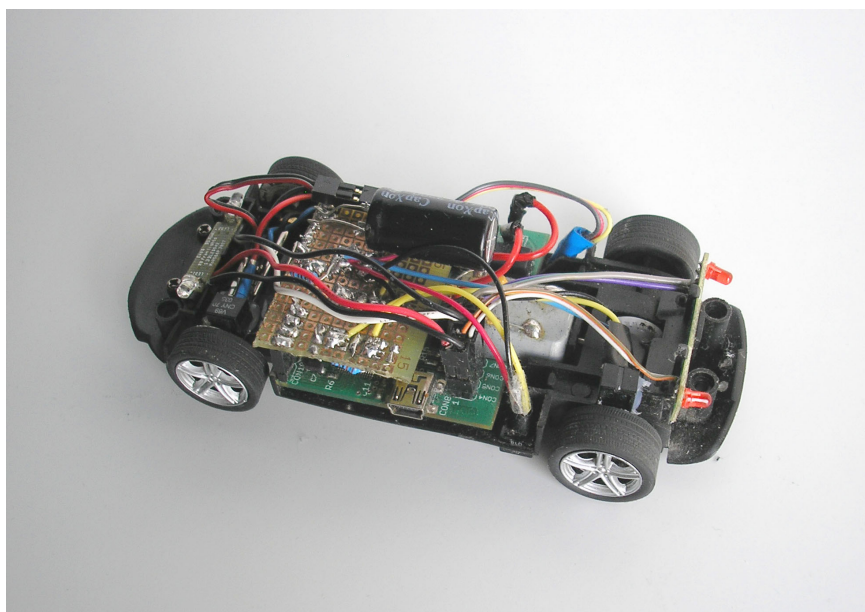


Obrázek 9: Schéma plošného spoje s obvodem pro tvarování signálu.

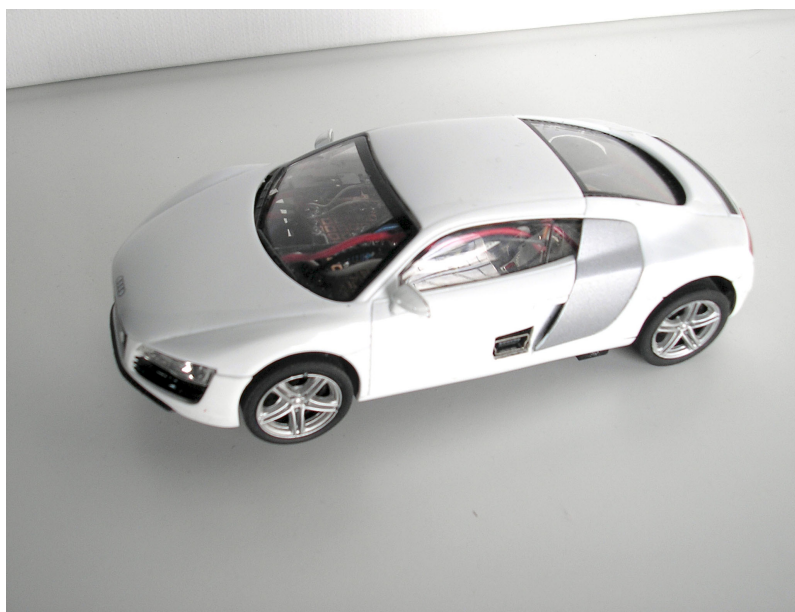




## A.2 Model vozu



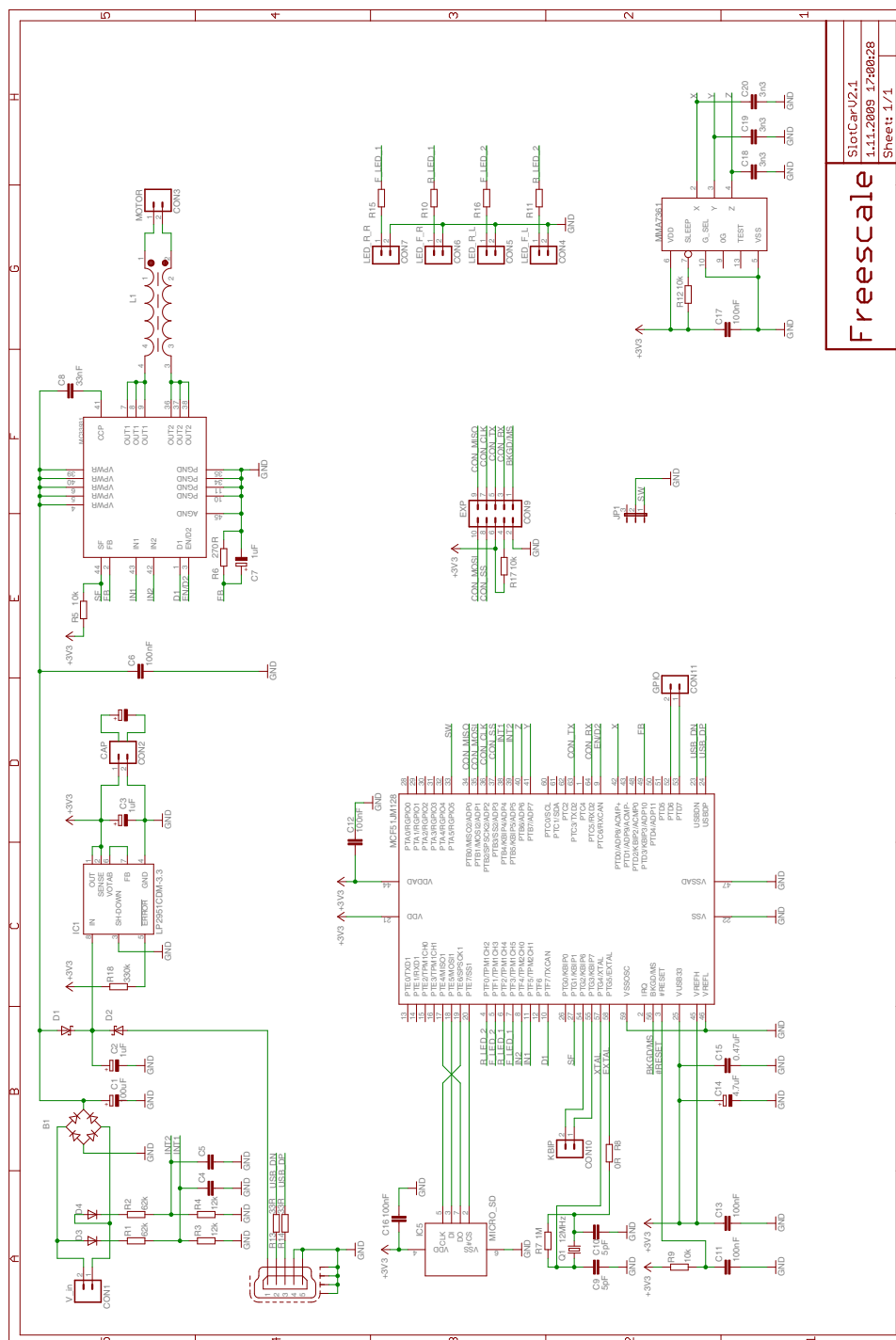
Obrázek 10: Pohled na vnitřní vybavení modelu.



Obrázek 11: Kompletní model vozu.



### A.3 Schéma řídicího obvodu



Obrázek 12: Schéma zapojení řídicího obvodu [5].



## A.4 Obsah přiloženého CD

- **Seznam adresářů**
  - **Bakalářská práce:** Text bakalářské práce ve formátu .pdf.
  - **Zdrojový kód CW:** Adresář obsahuje projekt v prostředí Codewarrior.
  - **Zdrojový kód:** Adresář obsahuje zdrojový kód ve formátu .rtf.
  - **Dokumenty:** Technické specifikace využitých součástek, výsledky soutěže 2011.
  - **Obrázky:** Použité obrázky.